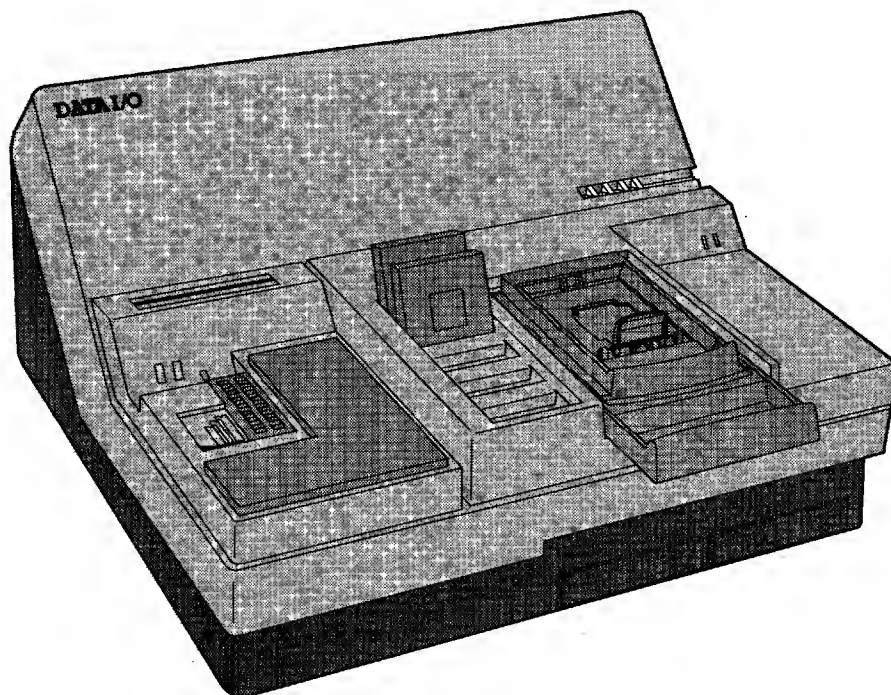


# UniSite™

---

UNIVERSAL PROGRAMMER



---

USER MANUAL

**DATA I/O**  
Corporation

981-0014

# UniSite<sup>TM</sup>

---

## Universal Programmer

### User Manual

June 1995

981-0014-013

Data I/O has made every attempt to ensure that the information in this document is accurate and complete. Data I/O assumes no liability for errors, or for any incidental, consequential, indirect or special damages, including, without limitation, loss of use, loss or alteration of data, delays, or lost profits or savings, arising from the use of this document or the product which it accompanies.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose without written permission from Data I/O.

Data I/O Corporation  
10525 Willows Road N.E., P.O. Box 97046  
Redmond, Washington 98073-9746 USA  
(206) 881-6444

#### Acknowledgments:

Data I/O is a registered trademark of Data I/O Corporation.  
UniSite, Site40, Site48, ChipSite, PinSite, SetSite, Keep Current,  
MatchBook, and HiTerm Terminal Emulator are trademarks of Data I/O Corporation.

Data I/O Corporation acknowledges the trademarks of other organizations for their respective products or services mentioned in this document.

© 1989-1995 Data I/O Corporation.  
All rights reserved

UniSite is protected under U.S. Patent numbers 4,837,653, 4,840,576, 5,033,084, 5,176,525, and 5,289,118. Other U.S. and Foreign Patents Pending.

# *Table of Contents*

---

## **Preface**

Data I/O Customer Support . . . . .	xiii
Technical Assistance . . . . .	xv
Calling . . . . .	xv
Sending a Fax . . . . .	xv
Sending E-mail . . . . .	xv
Using the BBS . . . . .	xv
Bulletin Board Service . . . . .	xvi
Warranty Information . . . . .	xvi
Keep Current Subscription Service . . . . .	xvii
Repair Service . . . . .	xvii
End User Registration and Address Change . . . . .	xvii
Typographic Conventions . . . . .	xviii

## **Safety Summaries**

English . . . . .	xix
German . . . . .	xxi
French . . . . .	xxiii
Italian . . . . .	xxv
Spanish . . . . .	xxvii

## **1. Introduction**

What Is UniSite? . . . . .	1-1
Contents of Package . . . . .	1-2
UniSite External Features . . . . .	1-3
The Front Panel . . . . .	1-3
The Back Panel . . . . .	1-4
Specifications . . . . .	1-5



Functional . . . . .	1-5
Power Requirements . . . . .	1-5
Physical and Environmental . . . . .	1-5
Safety . . . . .	1-6
Electrostatic Discharge (ESD) . . . . .	1-6
Certificate of RFI/EMI Compliance . . . . .	1-6
Calibration . . . . .	1-6
Options . . . . .	1-6
How Many Pin Driver Boards Do You Need? . . . . .	1-9

## 2. Setup and Installation

Before You Begin . . . . .	2-2
Licensing Agreement . . . . .	2-2
Definitions . . . . .	2-2
1. Install a Module . . . . .	2-3
Removing the Protective Cover . . . . .	2-3
Installing a Module . . . . .	2-3
Removing a Module . . . . .	2-4
2. Check the Line Voltage . . . . .	2-5
3. Choose Your Configuration . . . . .	2-5
4. Connect Your Equipment . . . . .	2-6
Connecting to a PC . . . . .	2-6
Installing HiTerm . . . . .	2-7
Connecting to a Host . . . . .	2-9
Connecting to a Terminal . . . . .	2-11
More About Cables . . . . .	2-13
Making Your Own Cable . . . . .	2-14
5. Insert the System Disk . . . . .	2-17
Software Version Compatibility . . . . .	2-19
6. Install the Base . . . . .	2-20
Installing a Base . . . . .	2-20
Removing a Base . . . . .	2-20
7. Power Up the Programmer . . . . .	2-21
Connecting the Power Cord . . . . .	2-21
Powering Up . . . . .	2-22
8. The Power-on Screen . . . . .	2-26
Selecting a New Terminal Type . . . . .	2-26
Changing the Default Terminal . . . . .	2-27
9. Insert Algorithm Disk . . . . .	2-28
10. Update System Software . . . . .	2-28
11. Set Up High Speed Download . . . . .	2-29
Why Should I Use High Speed Download? . . . . .	2-29

---

Configuring the Remote Port . . . . .	2-29
Changing the Powerup Defaults . . . . .	2-31
Using High Speed Download . . . . .	2-31
12. Install and Update the MSM (Mass Storage Module) . . . . .	2-32
Hardware Installation Instructions . . . . .	2-32
Software Installation and Update . . . . .	2-32
Bootting from the MSM . . . . .	2-33
Storage Capacity . . . . .	2-33
Storage Suggestions . . . . .	2-33
Limitations . . . . .	2-33
Backing up the MSM . . . . .	2-33
13. Install Devices . . . . .	2-34
Inserting a DIP Device . . . . .	2-34
Installing a MatchBook . . . . .	2-34
Inserting a PLCC or LCC Device into a MatchBook . . . . .	2-36
Inserting an SOIC Device into a MatchBook . . . . .	2-37
Installing a PGA Device in a PGA Base . . . . .	2-38
Removing a Device From a MatchBook . . . . .	2-39
Special Note About the Conductive Pad . . . . .	2-39
Maintenance . . . . .	2-40
Changing the Line Voltage . . . . .	2-40
Replacing the Line Fuse . . . . .	2-41
What to Do Next Time . . . . .	2-43

### 3. Quick Start

Before You Begin . . . . .	3-1
The UniSite Screen . . . . .	3-2
Moving the Cursor . . . . .	3-3
Selecting a Command . . . . .	3-3
Accessing Online Help . . . . .	3-3
Getting Help . . . . .	3-3
Accessing Online Help for System Messages . . . . .	3-4
Accessing Device-Specific Online Information . . . . .	3-4
Exiting Help . . . . .	3-5
Using Key Functions . . . . .	3-5

### 4. Tutorial

Before You Begin . . . . .	4-1
Should I Read This Chapter? . . . . .	4-1
What Will I Learn? . . . . .	4-2
Review of Programming Concepts . . . . .	4-2
Select Device . . . . .	4-2

Load Data . . . . .	4-2
Program Device . . . . .	4-2
Verify Device . . . . .	4-2
Session 1: Navigating Through the Menus . . . . .	4-3
Select a Menu Item . . . . .	4-3
Select Online Help . . . . .	4-4
Exit the Online Help . . . . .	4-5
Review . . . . .	4-5
Session 2: Selecting a Device . . . . .	4-6
Before You Begin . . . . .	4-6
Can I Use Another Device? . . . . .	4-6
Select a Manufacturer . . . . .	4-6
Select a Device Part Number . . . . .	4-8
Reviewing Your Options . . . . .	4-8
Accessing Device-specific Online Information . . . . .	4-10
About Family/Pinout Codes . . . . .	4-10
Review . . . . .	4-10
Session 3: Selecting a Keep Current Algorithm . . . . .	4-11
Before You Begin . . . . .	4-11
Can I Use Another Device? . . . . .	4-11
Select the Keep Current Option . . . . .	4-11
Select the Keep Current Algorithm . . . . .	4-13
Found It . . . . .	4-14
Keep Current Algorithms and Software Updates . . . . .	4-14
Special Note . . . . .	4-15
Session 4: Loading Data from a Device . . . . .	4-16
Before You Begin . . . . .	4-16
Can I Use Another Device? . . . . .	4-16
Select a Device . . . . .	4-16
Insert the Device . . . . .	4-17
Load the Data . . . . .	4-17
Parameter Screens . . . . .	4-18
Setting the Parameters . . . . .	4-19
Loading the Data . . . . .	4-19
Review . . . . .	4-19
Session 5: Loading Data from Disk . . . . .	4-20
Before You Begin . . . . .	4-20
Getting There . . . . .	4-20
Review . . . . .	4-22
Session 6: Selecting a Translation Format . . . . .	4-23
About Translation Formats . . . . .	4-23
Getting There . . . . .	4-23

Review . . . . .	4-23
Session 7: Loading Data from a PC Using HiTerm . . . . .	4-24
Before You Begin . . . . .	4-24
What to Transfer? . . . . .	4-24
Downloading the File . . . . .	4-26
After the Download . . . . .	4-27
Review . . . . .	4-27
Session 8: Loading Data from a Host . . . . .	4-28
Before You Begin . . . . .	4-28
What to Transfer? . . . . .	4-28
Downloading the File . . . . .	4-30
After the Download . . . . .	4-31
Review . . . . .	4-31
Session 9: Editing Data . . . . .	4-32
Before You Begin . . . . .	4-32
Moving Around . . . . .	4-33
Making Changes . . . . .	4-33
Restoring Your File . . . . .	4-34
Saving Your Edits . . . . .	4-34
Review . . . . .	4-34
Session 10: Programming a Memory Device . . . . .	4-35
Before You Begin . . . . .	4-35
Can I Use Another Device? . . . . .	4-35
Parameter Screens . . . . .	4-35
What Is the Difference? . . . . .	4-36
Set the Programming Parameters . . . . .	4-37
Programming the Device . . . . .	4-37
Review . . . . .	4-37
Session 11: Verifying a Device . . . . .	4-38
Before You Begin . . . . .	4-38
Can I Use Another Device? . . . . .	4-38
Parameter Screens . . . . .	4-38
Setting the Verify Parameters . . . . .	4-39
Verifying the Device . . . . .	4-39
Review . . . . .	4-40

## 5. Commands

Menu Organization . . . . .	5-1
Menu Maps . . . . .	5-1
Overwriting User RAM . . . . .	5-3
Factory Default Settings . . . . .	5-3
Select Device . . . . .	5-5

Before You Select a Device . . . . .	5-5
Selecting a Device . . . . .	5-6
Family/Pinout Code Selection . . . . .	5-9
After You Select a Device . . . . .	5-10
Cross Programming . . . . .	5-11
Quick Copy . . . . .	5-12
Load Device . . . . .	5-13
Load Logic Device . . . . .	5-13
Load Memory Device . . . . .	5-14
Program Device . . . . .	5-17
Program Logic Device . . . . .	5-17
Program Memory Device . . . . .	5-19
Enhanced Security Fuse Capability . . . . .	5-22
Verify Device . . . . .	5-23
Verify Logic Device . . . . .	5-23
Verify Memory Device . . . . .	5-24
More Commands . . . . .	5-27
Configure System . . . . .	5-28
Programmer ID . . . . .	5-42
Carrying a Configuration File Forward . . . . .	5-43
Keep Current . . . . .	5-46
Custom Menu Algs . . . . .	5-50
Mass Storage . . . . .	5-55
Device Checks . . . . .	5-56
Edit Data . . . . .	5-66
File Operations . . . . .	5-76
Job File . . . . .	5-85
Remote Control . . . . .	5-87
Self-test . . . . .	5-87
Transfer Data . . . . .	5-89
Yield Tally . . . . .	5-99

## **6. Computer Remote Control**

Which Driver to Use? . . . . .	6-1
System Setup . . . . .	6-2
Entering CRC Mode . . . . .	6-2
Exiting CRC Mode . . . . .	6-4
Halting CRC Operations . . . . .	6-4
CRC Default Settings . . . . .	6-5
CRC Commands . . . . .	6-6
CRC Command Summary . . . . .	6-6

## 7. Translation Formats

About this Chapter . . . . .	7-1
Instrument Control Codes . . . . .	7-3
General Notes . . . . .	7-4
ASCII Binary Format, Codes 01, 02 and 03 (or 05, 06 and 07) . . . . .	7-5
Texas Instruments SDSMAC Format (320), Code 04 . . . . .	7-7
The 5-Level BNPF Format, Codes 08 or 09 . . . . .	7-9
Formatted Binary format, Code 10 . . . . .	7-10
DEC Binary Format, Code 11 . . . . .	7-12
Spectrum Format, Codes 12 or 13 . . . . .	7-12
POF (Programmer Object File) Format, Code 14 . . . . .	7-13
Absolute Binary Format, Code 16 . . . . .	7-15
LOF Format, Code 17 . . . . .	7-15
LOF Field Syntax . . . . .	7-15
ASCII Octal and Hex Formats, Codes 30-37 and 50-58 . . . . .	7-17
RCA Cosmac Format, Code 70 . . . . .	7-19
Fairchild Fairbug, Code 80 . . . . .	7-20
MOS Technology Format, Code 81 . . . . .	7-21
Motorola EXORciser Format, Code 82 . . . . .	7-22
Intel Intellec 8/MDS Format, Code 83 . . . . .	7-23
Signetics Absolute Object Format, Code 85 . . . . .	7-24
Tektronix Hexadecimal Format, Code 86 . . . . .	7-25
Motorola EXORmacs Format, Code 87 . . . . .	7-26
Intel MCS-86 Hexadecimal Object, Code 88 . . . . .	7-27
Hewlett-Packard 64000 Absolute Format, Code 89 . . . . .	7-29
Texas Instruments SDSMAC Format, Code 90 . . . . .	7-31
JEDEC Format, Codes 91 and 92 . . . . .	7-32
Introduction . . . . .	7-32
BNF Rules and Standard Definitions . . . . .	7-32
JEDEC Full Format, Code 91 . . . . .	7-35
JEDEC Field Syntax . . . . .	7-36
Field Identifiers . . . . .	7-36
JEDEC Kernel Mode, Code 92 . . . . .	7-42
Extended Tektronix Hexadecimal Format, Code 94 . . . . .	7-43
Motorola 32-Bit Format, Code 95 . . . . .	7-44
Hewlett-Packard UNIX Format, Code 96 . . . . .	7-45
Intel OMF386 Format, Code 97 . . . . .	7-47
Intel OMF286 Format, Code 98 . . . . .	7-48
Intel Hex-32, Code 99 . . . . .	7-50
Highest I/O Addresses . . . . .	7-52

## 8. Messages

Message List . . . . .	8-1
------------------------	-----

## Glossary

## Index

## Figures

Figure 1-1. Contents of the UniSite Universal Programmer . . . . .	1-2
Figure 1-2. Front Panel Features . . . . .	1-3
Figure 1-3. Back Panel Features . . . . .	1-4
Figure 2-1. Installing a Module . . . . .	2-4
Figure 2-2. UniSite's Rear Panel . . . . .	2-5
Figure 2-3. Transparent Mode . . . . .	2-9
Figure 2-4. Transparent Mode . . . . .	2-11
Figure 2-5. Pin Designations for RS-232C Serial Port Connection . . . . .	2-14
Figure 2-6. Write Enabling a 3.5-inch Disk . . . . .	2-17
Figure 2-7. Inserting the System Disk . . . . .	2-18
Figure 2-8. Front Panel Indicators . . . . .	2-23
Figure 2-9. Typical Power On Screen . . . . .	2-26
Figure 2-10. Inserting a DIP Device into the DIP Base . . . . .	2-34
Figure 2-11. Inserting a MatchBook into the Base . . . . .	2-35
Figure 2-12. Closing the MatchBook . . . . .	2-35
Figure 2-13. Inserting a Device into the PLCC or LCC Base . . . . .	2-36
Figure 2-14. Inserting an SOIC Device . . . . .	2-37
Figure 2-15. Orienting a PGA Device in the PGA Base . . . . .	2-38
Figure 2-16. Removing the Voltage Selector Wheel . . . . .	2-40
Figure 2-17. Voltage Reading . . . . .	2-41
Figure 2-18. Opening the Fuse Holder . . . . .	2-41
Figure 2-19. Removing the Fuse Holder . . . . .	2-42
Figure 2-20. Inserting the Fuse Holder . . . . .	2-42
Figure 3-1. The Main Menu . . . . .	3-1
Figure 3-2. Areas of the UniSite Screen . . . . .	3-2
Figure 3-3. Areas of the Help Screen . . . . .	3-4
Figure 4-1. The More Commands Menu Screen . . . . .	4-3
Figure 4-2. The Self-test Screen . . . . .	4-4
Figure 4-3. Areas of the Help Screen . . . . .	4-4
Figure 4-4. The Device Manufacturer Selection Screen . . . . .	4-6
Figure 4-5. The Part Number Selection Screen . . . . .	4-8
Figure 4-6. The Device Manufacturer Selection Screen . . . . .	4-11
Figure 4-7. The Keep Current Part List Screen . . . . .	4-13
Figure 4-8. Inserting a DIP Device . . . . .	4-17

Figure 4-9. The Load Memory Device Parameter Screen . . . . .	4-17
Figure 4-10. The All Parameters Load Memory Screen . . . . .	4-18
Figure 4-11. The File Menu . . . . .	4-21
Figure 4-12. The Load File Dialog Screen . . . . .	4-21
Figure 4-13. The Translation Format Selection Screen . . . . .	4-23
Figure 4-14. The Download Data from Host Screen . . . . .	4-25
Figure 4-15. Transparent Mode . . . . .	4-28
Figure 4-16. The Download Data from Host Screen . . . . .	4-29
Figure 4-17. The Edit Programmer Memory Screen . . . . .	4-32
Figure 4-18. The Edit Screen . . . . .	4-32
Figure 4-19. The Program Memory Device Screen . . . . .	4-35
Figure 4-20. The All Parameters Screen . . . . .	4-36
Figure 4-21. The Programming Parameters for This Session . . . . .	4-37
Figure 4-22. The All Parameters Verify Screen . . . . .	4-38
Figure 4-23. The Non-default Verify Parameters Screen . . . . .	4-39
Figure 5-1. The Command Tree . . . . .	5-2
Figure 7-1. An Example of ASCII Binary Format . . . . .	7-5
Figure 7-2. An Example of TI SDSMAC Format . . . . .	7-7
Figure 7-3. An Example of Formatted Binary Format . . . . .	7-10
Figure 7-4. An Example of Formatted Binary Format . . . . .	7-11
Figure 7-5. An Example of Spectrum Format . . . . .	7-12
Figure 7-6. An Example of ASCII Octal and Hex Formats . . . . .	7-17
Figure 7-7. An Example of RCA Cosmac Format . . . . .	7-19
Figure 7-8. An Example of Fairchild Fairbug . . . . .	7-20
Figure 7-9. An Example of MOS Technology Format . . . . .	7-21
Figure 7-10. An Example of Motorola EXORciser Format . . . . .	7-22
Figure 7-11. An Example of Intel Intellec 8/MDS Format . . . . .	7-23
Figure 7-12. An Example of Signetics Absolute Object Format . . . . .	7-24
Figure 7-13. An Example of Tektronix Hex Format . . . . .	7-25
Figure 7-14. An Example of Motorola EXORmacs Format . . . . .	7-26
Figure 7-15. An Example of Intel MCS-86 Hex Object . . . . .	7-27
Figure 7-16. An Example of HP 64000 Absolute Format . . . . .	7-29
Figure 7-17. An Example of the TI SDSMAC Format . . . . .	7-31
Figure 7-18. An Example of JEDEC Full Format . . . . .	7-35
Figure 7-19. An Example of JEDEC Kernel Mode Format . . . . .	7-42
Figure 7-20. An Example of Tektronix Extended Format . . . . .	7-43
Figure 7-21. An Example of Motorola S3 Format . . . . .	7-44
Figure 7-22. Hewlett-Packard 64000 Unix Format . . . . .	7-46
Figure 7-23. A Sample of the Intel OMF286 Format . . . . .	7-48
Figure 7-24. A Close-up of the Intel OMF286 Format . . . . .	7-49
Figure 7-25. An Example of the Intel Hex-32 Format . . . . .	7-50





# *Preface*

---

The Preface includes details about contacting Data I/O for technical assistance, repair and warranty services, and Keep Current™ subscription service. The Preface also explains the Bulletin Board Service, typographic conventions and more.

---

## **Data I/O Customer Support**

### **United States**

For technical assistance, contact

**Data I/O Customer Resource Center**

Telephone: 800-247-5700

Fax: 206-869-2821

For repair or warranty service, contact

**Data I/O Central Dispatch**

Telephone: 800-735-6070

Fax: 206-881-0561

For Keep Current subscription service, contact

**Data I/O Sales**

Telephone: 800-332-8246

### **Canada**

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

**Data I/O Canada**

6725 Airport Road, Suite 102

Mississauga, Ontario, L4V 1V2

Telephone: 905-678-0761

Fax: 905-678-7306

## **United Kingdom**

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

**Data I/O Limited**  
660 Eskdale Road  
Winnersh, Wokingham  
Berkshire RG11 5TS  
Telephone: (0)1734-440011  
Fax: (0)1734-448700

## **Japan**

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

**Data I/O Japan**  
Osaki CN Building 2F  
5-10-10, Osaki Shinagawa-Ku  
Tokyo 141  
Telephone: 3-3779-2152  
Fax: 3-3779-2207 (Operations)  
3-3779-2203 (Other)

## **Germany**

For technical assistance, repair, warranty service, or Keep Current subscription service, contact

**Data I/O GmbH**  
Lochhamer Schlag 5a  
82166 Gräfelfing  
Telephone: 089-858580  
Fax: 089-8585810

## **Other European Countries**

For technical assistance, repair, warranty service, or Keep Current subscription service, contact the office below and ask for the number of your local Data I/O representative.

**Data I/O Europe**  
660 Eskdale Road  
Winnersh, Wokingham  
Berkshire RG11 5TS  
United Kingdom  
Telephone: 44-(0)1734-448899  
Fax: 44-(0)1734-440070

## **Other Countries Worldwide**

For technical assistance, repair, warranty service, or Keep Current subscription service, contact the office below and ask for the number of your local Data I/O representative.

**Data I/O Intercontinental**  
10525 Willows Road N.E.  
P.O. Box 97046  
Redmond, WA USA 98073-9746  
Telephone: 206-881-6444  
Fax: 206-882-1043  
Telex: 4740166

---

## Technical Assistance

You may contact Data I/O for technical assistance by calling, sending a fax or electronic mail (e-mail), or using the Bulletin Board Service (BBS). To help us give you quick and accurate assistance, please provide the following information:

- Product version number
- Product serial number (if available)
- Detailed description of the problem you are experiencing
- Error messages (if any)
- Device manufacturer and part number (if device-related)

### Calling

Call the appropriate Data I/O Customer Support number listed at the front of the Preface. When you call, please be at your programmer or computer, have the product manual nearby, and be ready to provide the information listed above.

### Sending a Fax

Fax the information listed above with your name, telephone number, and address to the appropriate Data I/O Customer Support fax number listed at the front of the Preface.

### Sending E-mail

To reach Data I/O via e-mail, send a message including your name, telephone number, e-mail address, and the information listed above to one of the following addresses:

`techhelp`

or

`{apple|decwrl|rutgers|gatech|uunet}!pilchuck!techhelp`

---

*Note: Select one of the five addresses listed above in braces. For example, you might send e-mail to the following address: `uunet!pilchuck!techhelp`. See your system administrator if you need more information on which address to use.*

### Using the BBS

To reach Data I/O via the BBS, include your name, telephone number, e-mail address, and the information listed above in a message, and send it to the BBS as described in the following section.

---

## Bulletin Board Service

From the Data I/O Bulletin Board System (BBS) you can obtain a wide range of information on Data I/O products, including current product descriptions, new revision information, technical support information, application notes, and other miscellaneous information. Using the BBS, you can access device support information, request support for a particular device, and leave messages for the BBS system operator, Customer Support personnel, or other customers. The BBS also includes many downloadable DOS utilities. Multiple lines are available, all supporting 1200/2400/9600/19200 baud, with U.S. Robotics Dual/HST V.32bis/V.42bis modems. The modems are set to 8 data bits, 1 stop bit, and no parity. Online help files provide more information about the BBS and its capabilities. BBS numbers for all countries are as follows:

Country	BBS Number
Belgium	+32-(0)1-6380731
Canada	+1-905-678-0547
France	+33-(0)13-9562699
Germany	+49-(0)89-8585833
Japan	+81-33779-2240
Netherlands	+31-(0)40-582424
Norway	+47-(0)66-780445
Sweden	+46-(0)8-7391037
Switzerland	+41-(0)1-313-0658
United Kingdom	+44-(0)1734-448862
United States	+1-206-882-3211

---

## Warranty Information

Data I/O Corporation warrants this product against defects in materials and workmanship at the time of delivery and thereafter for a period of one (1) year. The foregoing warranty and the manufacturers' warranties, if any, are in lieu of all other warranties, expressed, implied or arising under law, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Data I/O maintains customer service offices throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. For warranty service, contact Data I/O Customer Support at the numbers listed at the front of the Preface.

---

## Keep Current Subscription Service

Data I/O offers a one-year renewable subscription to keep your product and documentation up-to-date with the latest features and device support. This subscription, called Keep Current subscription service, also incorporates manufacturer-recommended changes to existing device support to maintain optimum yields, throughput, and long-term reliability. For more information, or to order Keep Current subscription service, contact Data I/O Customer Support at the numbers listed at the front of the Preface.

---

## Repair Service

After the warranty period expires, repair services are available at Data I/O Service Centers worldwide. Single instance repairs and fixed price annual agreements that cover all parts and labor needed to correct normal malfunctions are also available. The annual agreements include semiannual performance certification. For more information, or to order a Repair Service Agreement, contact Data I/O Customer Support at the numbers listed at the front of the Preface.

---

## End User Registration and Address Change

If the end user for this product or your address has changed since the Registration Card was mailed, please notify Data I/O Customer Support at the numbers listed at the front of the Preface. This ensures that you receive information about product enhancements. Be sure to include the product serial number, if available.

## Typographic Conventions

Throughout this manual different typographic conventions represent different cases of input and output.

### Keys

Keys appear in boxes (for example, **Q**) or as bolded text. The **Enter** key (or on some keyboards, the **Return** key) is represented by this symbol: **↵**.

### Key Combinations

An instruction for pressing two keys at once, such as **^Z** (Control and Z), is represented by two key boxes separated by a plus, such as **Ctrl + Z**. A key combination like **Esc Ctrl + T** means press and release **Esc**, then press **Ctrl** and **T** at the same time.

### Variable Input

Variable input is italicized and should be replaced with the requested information. For example, "enter *copy filename.hex*" means type **copy** just as you see it and replace *filename.hex* with the name of your file.

### Displayed Text

Text displayed on an LCD or screen appears in a typewriter-like typeface.

You will see this text displayed on the screen.

# *Safety Summary*

---

General safety information for operating personnel is contained in this summary. In addition, specific **WARNINGS** and **CAUTIONS** appear throughout this manual where they apply and are not included in this summary.

## **Antistatic Wrist Strap**

To avoid electric shock, the antistatic wrist strap must contain a 1M $\Omega$  (minimum) to 10M $\Omega$  (maximum) isolating resistor.

## **Definitions**

**WARNING** statements identify conditions or practices that could result in personal injury or loss of life. **CAUTION** statements identify conditions or practices that could result in damage to equipment or other property.

## **Fuse Replacement**

For continued protection against the possibility of fire, replace the fuse only with a fuse of the specified voltage, current and type ratings.

## **Grounding the Product**

The product is grounded through the grounding conductor of the power cord. To avoid electric shock, plug the power cord into a properly wired and grounded receptacle only. Grounding this equipment is essential for its safe operation.

## **Power Cord**

Use only the power cord specified for your equipment.

## **Power Source**

To avoid damage, operate the equipment only within specified line (ac) voltage.

## **Servicing**

To reduce the risk of electric shock, perform only the servicing described in this manual.



## Symbols



This symbol indicates that the user should consult the manual for further detail.



This symbol stands for Vac, for example, 120 V  $\sim$  = 120 Vac.



This symbol denotes a fuse rating for a user-replaceable fuse.



This symbol denotes the protective ground connection.



This symbol denotes a ground connection for a signal or for an antistatic wrist strap with impedance of 1M $\Omega$  (minimum) to 10M $\Omega$  (maximum).

## Certificate of RFI/EMI Compliance

Data I/O certifies that this product complies with the Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) requirements of EN55022 Class A and EN50082-1 Criteria C as called out in 89/336/EEC, the EMC Directive for the European Community.

**CE** - EC conformity mark

Upon request, Data I/O will supply a copy of the Declaration of Conformity, submitted per 89/336/EEC.

# **Zusammenfassende Sicherheitsinformationen**

---

Diese Zusammenfassung enthält allgemeine Sicherheitsinformationen für das Bedienerpersonal. Zusätzlich erscheinen, wenn zutreffend, ausdrückliche Hinweise (ACHTUNG!, VORSICHT!) im Verlauf des Textes. Diese Hinweise werden in dieser Zusammenfassung nicht wiederholt.

<b>Antistatik-Armband</b>	Zum Schutz gegen Stromschläge muß das Antistatik-Armband einen Isolierwiderstand von minimal 1M $\Omega$ und maximal 10M $\Omega$ enthalten.
<b>Definitionen</b>	Mit <b>ACHTUNG!</b> überschriebene Hinweise dienen zur Identifizierung und Warnung vor Zuständen oder Vorgängen, die Verletzungen oder Tod herbeiführen können. <b>VORSICHT!</b> dient zum Hinweis auf Zustände und Schritte, die zu Geräte-oder andersartigen Sachschäden führen können.
<b>Ersetzen von Sicherungen</b>	Ersetzen Sie zu Ihrem Schutz gegen Brandgefahr eine durchgebrannte Sicherung nur mit einer Sicherung der angegebenen Nennspannung, Stromart und Typenbestimmung.
<b>Erdung des Gerätes</b>	Das Gerät wird durch den dritten Leiter des Netzkabel geerdet. Stecken Sie die Netzschnur zur Vermeidung von Stromschlägen nur in eine geerdete Steckdose. Richtige Erdung ist für den problemfreien Betrieb dieses Gerätes unerlässlich.
<b>Netzkabel</b>	Verwenden Sie nur die für dieses Gerät vorgesehene Netzkabel.
<b>Stromquelle</b>	Vermeiden Sie Beschädigungen des Gerätes durch den Betrieb an der vorgeschriebenen Wechselspannung.
<b>Wartung/Reparatur</b>	Führen Sie zum Vermeiden von Stromschlägen nur die in diesem Handbuch erwähnten Wartungsarbeiten durch.

## Symbole



Dieses Symbol bedeutet, daß das Handbuch weitere dem Bediener hilfreiche Hinweise enthält.



Dieses Symbol bedeutet VAC (Volt Wechselstrom); z.B. 120V  $\sim$  = 120 VAC.



Dieses Symbol bezeichnet Sicherungsdaten für vom Bediener auszuwechselnde Sicherungen.



Dieses Symbol bezeichnet eine Schutz Erde-Verbindung.



Dieses Symbol bezeichnet eine Masseverbindung für ein Signal oder ein Antistatik-Armband mit einer Impedanz von 1M $\Omega$  (min) bis 10M $\Omega$  (max).

## Certificate of RFI/EMI Compliance

Data I/O certifies that this product complies with the Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) requirements of EN55022 Class A and EN50082-1 Criteria C as called out in 89/336/EEC, the EMC Directive for the European Community.

**CE** - EC conformity mark

Upon request, Data I/O will supply a copy of the Declaration of Conformity, submitted per 89/336/EEC.

# **Résumé des consignes de sécurité**

---

Ce résumé comprend les informations relatives à la sécurité pour les opérateurs. De plus, tout au long de ce manuel, on retrouve aux endroits appropriés, des **MISES EN GARDE** et des **AVERTISSEMENTS** spécifiques qui ne sont pas inclus dans ce résumé.

<b>Bracelet antistatique</b>	Afin d'éviter tout choc électrique, le bracelet antistatique doit renfermer un résistor de 1M $\Omega$ (minimum) à 10M $\Omega$ (maximum).
<b>Définitions</b>	Les indications de <b>MISE EN GARDE</b> signalent les conditions ou pratiques qui pourraient causer des blessures corporelles ou la mort. Les indications d' <b>AVERTISSEMENTS</b> signalent les conditions ou pratiques qui pourraient endommager l'équipement ou entraîner d'autres dommages matériels.
<b>Remplacement du fusible</b>	Pour assurer une protection continue contre les risques d'incendie, il faut remplacer le fusible uniquement avec un fusible de voltage, courant et type spécifiés.
<b>Mise à la terre du produit</b>	Le produit est mis à la terre par l'entremise de la borne de mise à la terre du cordon d'alimentation. Pour éviter tout choc électrique, il faut brancher le cordon d'alimentation uniquement dans un réceptacle mis à la terre correctement et dont les fils ont été rattachés correctement. Il est essentiel de mettre cet appareil à la terre pour qu'il puisse fonctionner sans danger.
<b>Cordon d'alimentation</b>	N'utiliser que le cordon d'alimentation spécifié pour votre appareil.
<b>Source d'alimentation</b>	Pour éviter d'endommager l'appareil, il faut respecter la tension (ca) spécifiée.
<b>Service</b>	Afin de réduire les risques de choc électrique, il faut s'en tenir aux opérations d'entretien et de réparation spécifiées dans ce manuel.

## Symboles



Ce symbole indique que l'utilisateur doit consulter le manuel pour obtenir de plus amples détails.



Ce symbole représente le voltage en courant alternatif V ca, par exemple, 120 V  $\sim$  = 120 V ca.



Ce symbole indique la valeur nominale d'un fusible remplaçable par l'utilisateur.



Ce symbole indique la connexion d'isolation à la masse.



Ce symbole indique une connexion de masse pour un signal ou un bracelet antistatique avec une impédance de 1M $\Omega$  (minimum) 10M $\Omega$  (maximum).

## Certificate of RFI/EMI Compliance

Data I/O certifies that this product complies with the Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) requirements of EN55022 Class A and EN50082-1 Criteria C as called out in 89/336/EEC, the EMC Directive for the European Community.

**CE** - EC conformity mark

Upon request, Data I/O will supply a copy of the Declaration of Conformity, submitted per 89/336/EEC.

# *Riepilogo di sicurezza*

---

Questo riepilogo contiene informazioni di sicurezza per il personale addetto alle operazioni. Inoltre, specifiche note di **ATTENZIONE** e di **AVVISO** relative al contesto fanno parte di questo manuale e non sono state ripetute in questo riepilogo.

**Cinghia antistatica da polso**

Per evitare le scosse elettriche, la cinghia antistatica da polso deve contenere un resistore di isolamento da  $1\text{M}\Omega$  (minimo) a  $10\text{M}\Omega$  (massimo).

**Definizioni**

Le note di **ATTENZIONE** identificano condizioni o procedure che potrebbero causare infortuni personali o decessi. Le note di **AVVISO** identificano condizioni o procedure che potrebbero causare danni all'equipaggiamento o ad altra proprietà.

**Sostituzione dei fusibili**

Per una continua protezione contro l'eventualità di incendi, sostituire il fusibile solo con un fusibile dai valori nominali di tensione, corrente e tipo specificati.

**Messa a terra del prodotto**

Il prodotto viene messo a terra tramite il conduttore della messa a terra del cavo elettrico. Per evitare scosse elettriche, innestare il cavo elettrico in una presa correttamente cablata e messa a terra. La messa a terra di questo equipaggiamento è essenziale per un funzionamento sicuro.

**Cavo elettrico**

Usare solo il cavo elettrico specificato per l'equipaggiamento.

**Fonte di alimentazione**

Per evitare danni, operare l'equipaggiamento solo entro la tensione (ca) di linea specificata.

**Manutenzione**

Per ridurre il rischio di scossa elettrica, svolgere solo la manutenzione descritta in questo manuale.

## Simboli



Questo simbolo indica che l'utente deve consultare il manuale per ulteriori dettagli.



Questo simbolo indica Vca, ad esempio,  $120V \sim = 120 V_{ca}$ .



Questo simbolo indica la capacità nominale di un fusibile che può venire sostituito dall'utente.



Questo simbolo contrassegna la messa a terra di protezione.



Questo simbolo contrassegna una messa a terra per un segnale o per un cinturino da polso antistatico con impedenza compresa tra  $1M\Omega$  (minimo) e  $10M\Omega$  (massimo).

## Certificate of RFI/EMI Compliance

Data I/O certifies that this product complies with the Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) requirements of EN55022 Class A and EN50082-1 Criteria C as called out in 89/336/EEC, the EMC Directive for the European Community.

**CE** - EC conformity mark

Upon request, Data I/O will supply a copy of the Declaration of Conformity, submitted per 89/336/EEC.

# *Resumen de seguridad*

---

En este resumen se proporciona información general sobre seguridad para el personal operativo. Además, aparecen notas de **ADVERTENCIA** y **CUIDADO** por todo el manual, donde son apropiadas y no se incluyen en este resumen.

## **Muñequera antiestática**

Para evitar descargas eléctricas, la muñequera antiestática debe contener un resistor aislante de 1 M $\Omega$  (como mínimo) a 10 M $\Omega$  (como máximo).

## **Definiciones**

Las notas de **ADVERTENCIA** identifican condiciones o prácticas que pudieran dar como resultado lesiones personales o pérdida de la vida. Las notas de **PRECAUCION** identifican condiciones o prácticas que pudieran dar como resultado daños en equipos u otras propiedades.

## **Reemplazo de fusibles**

Para tener protección continua contra las posibilidades de que se produzcan incendios, reemplace los fusibles sólo con otros del tipo, el voltaje y la corriente que se especifican.

## **Conexión a tierra del producto**

El producto se conecta a tierra por medio del conductor de masa del cable de alimentación. Para evitar descargas eléctricas, enchufe el cable de alimentación en un receptáculo alambrado y conectado a tierra de modo correcto.

## **Cable de alimentación**

Use sólo el cable de alimentación especificado para el equipo de que se trate.

## **Fuente de alimentación**

Para evitar daños, haga funcionar el equipo sólo dentro de los voltajes de línea especificados (de ca).

## **Servicios**

Para reducir los riesgos de que se produzcan descargas eléctricas, lleve a cabo sólo los servicios descritos en este manual.




## Símbolos



Este símbolo indica que el usuario debería consultar el manual para obtener más detalles.



Este símbolo representa V ca. Por ejemplo, 120 V  = 120 V ca.



Este símbolo denota un valor nominal para un fusible reemplazable por el usuario.



Este símbolo indica la conexión a tierra de protección.



Este símbolo equivale a una conexión a tierra para una señal o una banda pulsera de antiestática con una impedancia de 1M $\Omega$  (mínima) a 10M $\Omega$  (máxima).

## Certificate of RFI/EMI Compliance

Data I/O certifies that this product complies with the Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) requirements of EN55022 Class A and EN50082-1 Criteria C as called out in 89/336/EEC, the EMC Directive for the European Community.

**CE** - EC conformity mark

Upon request, Data I/O will supply a copy of the Declaration of Conformity, submitted per 89/336/EEC.

# 1 *Introduction*

---

---

## What Is UniSite?

The UniSite™ Universal Programmer is a tool for programming, verifying, and testing all programmable device technologies and packaging. The many facets of UniSite allow it to address a variety of needs in both engineering and manufacturing environments.

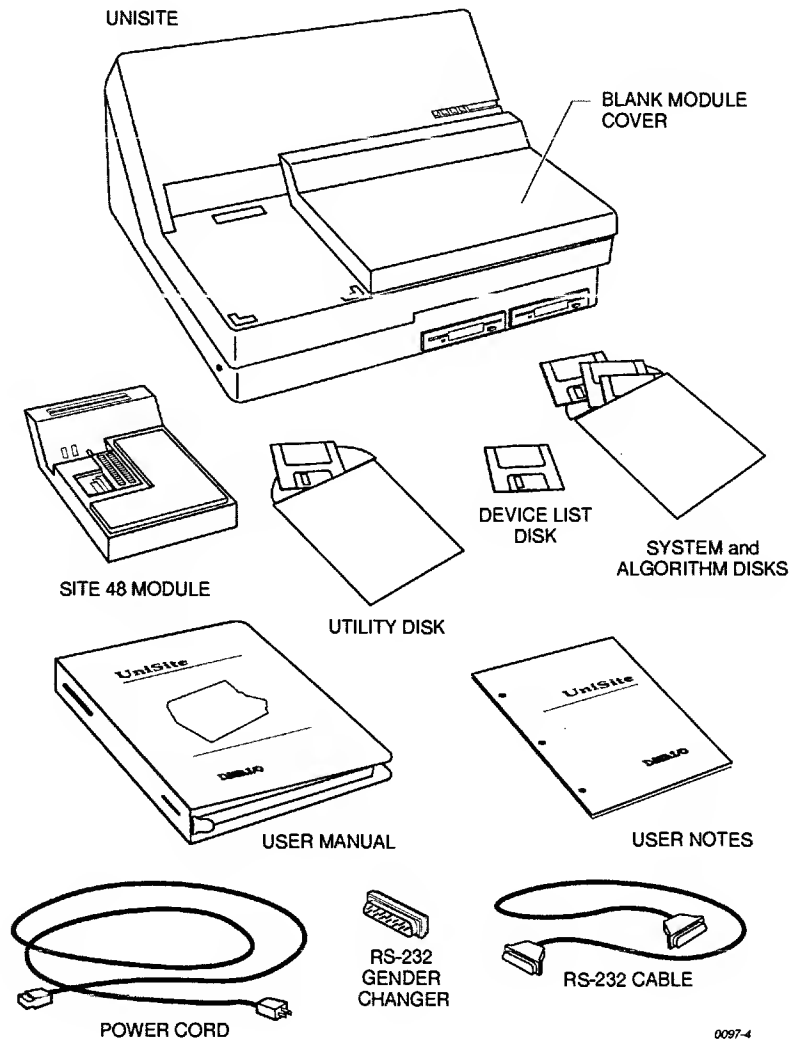
The basic UniSite comes with 1 MB of memory and support for DIP devices up to and including 40 pins. Options are available for support of devices in excess of 100 pins, in many package types, including DIP, PLCC, LCC, QFP, PGA, and SOIC.

Other options available for UniSite include a gang/set programming module for EEPROMs and EPROMs, a gang programming module for FPGAs, and an interface for automated device handlers.

## Contents of Package

Figure 1-1 illustrates the contents of the UniSite system. You should check the contents of your UniSite against the items shown in Figure 1-1.

**Figure 1-1**  
*Contents of the UniSite Universal Programmer*



*Note: You will not receive a blank module cover if you ordered a large module, such as PinSite or SetSite, when you ordered your UniSite.*

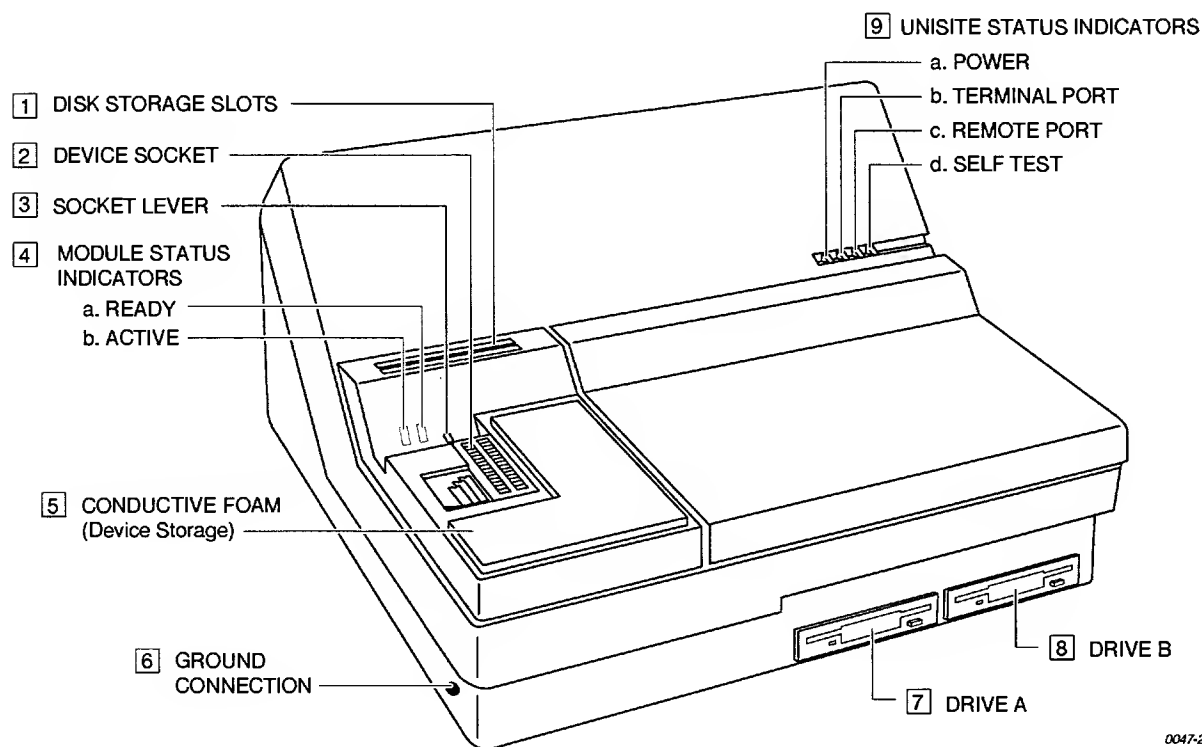
## UniSite External Features

### The Front Panel

The front panel features of UniSite are shown in Figure 1-2.

**Figure 1-2**

*Front Panel Features*



0047-2

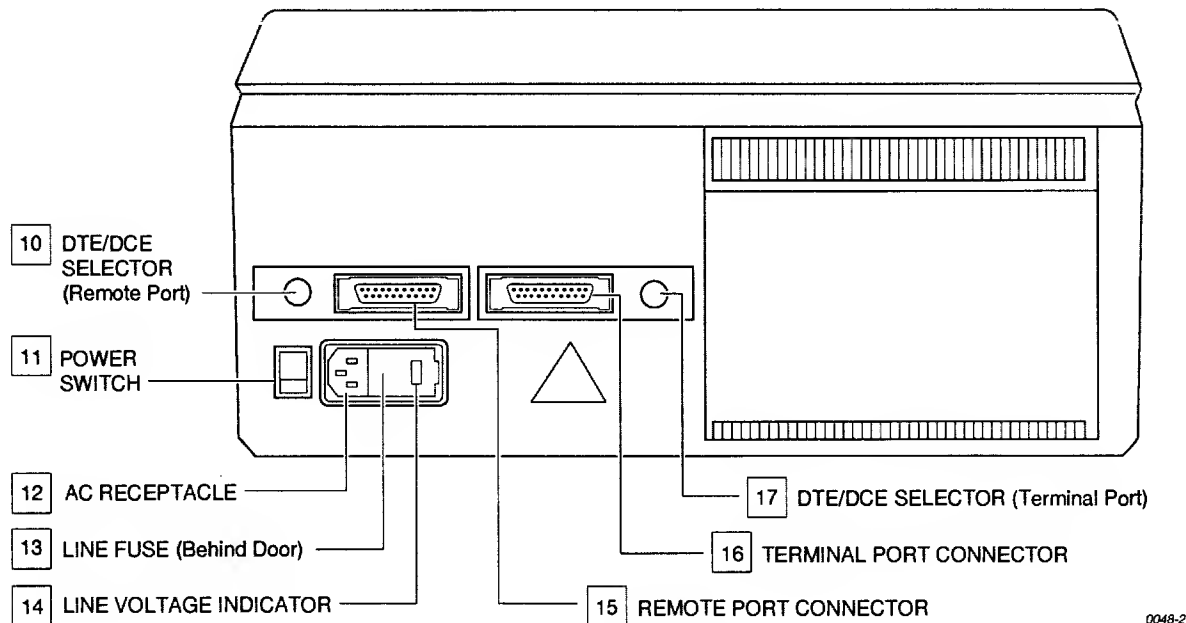
1. **Disk Storage Slots** — A convenient place to store two data disks.
2. **Device Socket** — Holds the device to be programmed or the master device to be read.
3. **Socket Lever** — Locks the device into the device socket.
4. **Module Status Indicators** — Provides information about the operational status of the module.
  - a. **ACTIVE Indicator** — This lamp is lit when a device-related operation is in progress.
  - b. **READY Indicator** — This lamp is lit when the device socket is ready to accept a device.
5. **Conductive Foam** — Provides a convenient, safe place to store static-sensitive devices while using UniSite.
6. **Ground Connection** — Connect an antistatic wrist strap here.
7. **Drive A** — The main disk drive. Insert the System disk here.
8. **Drive B** — The auxiliary disk drive. Insert an Algorithm disk here.

9. **UniSite Status Indicators** — These indicators provide information about UniSite's operational status:
  - a. **Power Indicator** — This lamp is lit when the power is on.
  - b. **Terminal Port Indicator** — This lamp is lit when equipment is connected properly to the Terminal port.
  - c. **Remote Port Indicator** — This lamp is lit when equipment is connected properly to the Remote port.
  - d. **Self-test Indicator** — This lamp is lit when UniSite is performing a self-test.

## The Back Panel

The back panel features of UniSite are shown in Figure 1-3.

**Figure 1-3**  
*Back Panel Features*



10. **Remote Port DTE/DCE Selector** — Configures the Remote port as Data Terminal Equipment or Data Communication Equipment.
11. **Power Switch** — Applies AC power to UniSite.
12. **AC Receptacle** — Connects UniSite to AC power.
13. **Line Fuse** — Houses the line fuse.
14. **Line Voltage Indicator** — Shows UniSite's line voltage.
15. **Remote Port** — Connects UniSite to PCs/workstations/terminals/file servers/etc.
16. **Terminal Port Connector** — Connects UniSite to PCs/workstations/terminals/file servers/etc.
17. **Terminal Port DTE/DCE Selector** — Configures the Terminal port as Data Terminal Equipment or Data Communication Equipment.

## Specifications

<b>Functional</b>	<b>RAM</b>	1 MB standard (up to 8 MB optional)
	<b>Disk Format</b>	Double-sided, double-density 3.5-inch disk with 135 tracks per inch. 720 KB formatted.
	<b>Controller</b>	Motorola 68000 16-bit microprocessor.
	<b>Terminal Support</b>	Interfaces with ANSI 3.64 compatible terminals, IBM PCs and compatibles running a terminal emulator program, and many popular ASCII terminals.
	<b>Communication Standard</b>	RS-232C.
	<b>Data transfer rate</b>	110 to 19.2 kilobaud (up to 115.2 kilobaud using HiTerm).
<b>Power Requirements</b>	<b>Operating Voltages</b>	100 to 120V ac $\pm 10\%$ or 220 - 240V ac $\pm 10\%$
	<b>Frequency Range</b>	48 to 63 Hz
	<b>Power Consumption</b>	500 VA maximum
	<b>Input Current</b>	5A maximum
	<b>Fuse Ratings</b>	250V/6A/F (Fast Blow)
<b>Physical and Environmental</b>	<b>Dimensions</b>	18.06h x 43.48w x 36.20d cm 7.11h x 17.12 w x 14.25d in.
	<b>Weight</b>	9.1 kg (20 lb)
	<b>Shipping Weight</b>	11.4 kg (25 lb)
	<b>Temperature</b>	Operating: +10° to +40°C (+50° to +105°F) Storage: +4° to +50°C (+40° to + 122°F) Transportation: -40° to +55°C (-40° to +130°F)
	<b>Relative Humidity</b>	Operating: 20 to 80% noncondensing Storage: 10 to 90% noncondensing
	<b>Altitude</b>	Operating: To 5,000 meters Storage: To 15,000 meters

## Safety

UniSite is certified by UL, CSA, and TUV to comply with the following safety standards:

Underwriters Laboratories — UL 1950

Canadian Standards Association — CSA C22.2 NO. 231

International Electrotechnical Commission — IEC 348 and IEC 1010-1

Verband Deutscher Electrotechniker — VDE 0411

## Electrostatic Discharge (ESD)

IEC 801-2 ( $\pm 8$  kV)

## Certificate of RFI/EMI Compliance

Data I/O certifies that this product complies with the Radio Frequency Interference (RFI) and Electromagnetic Interference (EMI) requirements of EN55022 Class A and EN50082-1 Criteria C as called out in 89/336/EEC, the EMC Directive for the European Community.

**CE** - EC conformity mark

Upon request, Data I/O will supply a copy of the Declaration of Conformity, submitted per 89/336/EEC.

## Calibration

UniSite verifies internal voltages every time it is powered up and every time a complete self-test is run. The voltage verification is performed by software and is compared to a laser-trimmed voltage reference. Data I/O recommends that you cycle power AND run a complete self-test cycle at least every three months. See the *UniSite Maintenance Manual* for more information on checking the reference voltages and the master clock.

To ensure that your UniSite continues to meet product performance specifications, Data I/O recommends that your programmer be returned to an authorized Data I/O Service Center every twelve months for a complete performance evaluation.

---

## Options

The items listed below complement the UniSite Universal Programmer. For more information, or to order an item below, contact Data I/O Customer Support as listed in the Preface.

### Keep Current. Subscription Service

Data I/O offers a one-year subscription to keep your programmer and documentation up-to-date with the latest features and device support. This subscription also incorporates manufacturer-recommended changes to existing device support to maintain optimum yields, throughput, and long-term reliability.

In addition, you receive immediate access to new and updated programming algorithms via our Keep Current Bulletin Board Service (BBS) — before the algorithms are available in the update kit.

For more information, see the Keep Current documentation located behind the Keep Current tab, or contact Data I/O Customer Support.

<b>RAM Upgrades</b>	Expands system RAM. Contact Data I/O Customer Support for more information on RAM upgrades.
<b>PinSite</b>	Expands device support to include several different package types, including PGA, PLCC, LCC, and SOIC. With the additional pin drivers located onboard PinSite, UniSite can support devices up to 84 pins. Uses MatchBook™ Device Carriers to improve throughput and reduce damage that may be caused by traditional sockets for surface-mount devices.
<b>PPI Base and Adapters</b>	A PinSite Base and Adapter combination that expands support to include devices with more than 84 pins and in additional package styles, such as SDIP, QFP, and memory cards.
<b>SetSite</b>	Provides set and gang programming support for up to eight E/EPROMs. Accommodates skinny or wide DIP EEPROMs and EEPROMs up to 40 pins.
<b>Site48HS</b>	Replaces Site40HS on existing UniSites and increases support for DIP devices to 48 pins.
<b>Second Disk Drive</b>	A second 3.5-inch disk drive for UniSite. Adding a second disk drive helps eliminate disk swaps and helps speed up device operations, such as programming and verifying. The disk drive does not require factory installation and can be installed on site.



<b>USM-340 (Actel FPGA Gang Module)</b>	Provides gang programming for up to eight Actel FPGAs. Eight 68-pin PLCC devices can be programmed simultaneously. Requires a minimum of 1 MB RAM in UniSite to program the Actel 1010. For larger devices, such as the Actel 1020, a minimum of 4 MB RAM is required.
<b>Pin Driver</b>	Provides additional support for devices with higher pin counts. Each pin driver board provides additional support for 4 pins. For example, if your UniSite has 10 pin drivers, you can program devices with up to 40 pins. If you want to program 48-pin devices, you will need to add two pin driver boards to your UniSite.
<b>Maintenance Manual</b>	Contains information on the following: disassembly/reassembly, circuit description, maintenance, and troubleshooting.
<b>Cable Set</b>	Includes an additional RS-232C cable and gender changer.
<b>Carrying Case</b>	Protects UniSite from damage during travel. The custom designed, soft-sided carrying case holds UniSite, the Algorithm disk, the System disk, and this manual. The Carrying Case is ideal if you will be using UniSite in the field.
<b>TaskLink™</b>	PC/Programmer Interface Software designed for use with UniSite. TaskLink runs on an IBM PC (or compatible) and allows you to control UniSite from a personal computer for streamlined and enhanced programming operations. TaskLink features automatic programming file configuration, full-screen editing, and error-logging. TaskLink also features a new windowed interface, extensive online context-sensitive help, and full mouse support.

## How Many Pin Driver Boards Do You Need?

The following table shows the minimum number of pin driver boards needed to program a device with a given number of pins. Each pin driver board can drive four device pins. Depending on how the pins on the device are mapped out, the number of pin driver boards required is sometimes greater than the number of device pins divided by four. For example, using Site 40 with a 32-pin device requires 9 pin driver boards instead of the theoretical minimum of 8 pin driver boards.

*Note: The information in the following table applies to UniSites running system software version 3.60 and later.*

Pin driver boards are easily installed and can be ordered by contacting your nearest Data I/O representative as listed in the Preface.

Number of Device Pins	Site 40 or HI-QA-DIP	Site 48HS	SetSite	ChipSite or HI-QA-SMT	PLCC, LCC, SOIC (PinSite)	PGA (PinSite)
8	7	7	10	7	7	n/a
14	7	7	10	7	7	n/a
16	7	7	10	7	7	n/a
18	7	7	10	7	7	n/a
20	7	7	10	7	7	n/a
24	7	7	10	7	7	n/a
28	7	7	10	7	7	17 *
32	9	8	10	9	8	n/a
40	10	10	10	n/a	10	n/a
44	n/a	11	n/a	12	11	17
48	n/a	12	n/a	n/a	12	n/a
52	n/a	n/a	n/a	15	13	n/a
68	n/a	n/a	n/a	17	17	17
84	n/a	n/a	n/a	n/a	17	17
>84	n/a	n/a	n/a	n/a	n/a	17

When you program devices with a PPI base, we recommend that you install 17 boards, although the requirements vary among adapters. For the exact number of boards you need with a specific PPI base, call Data I/O Customer Support as listed in the Preface.

\* Requires the PGA Base, PSBASE-0402. Contact Data I/O Customer Support as listed in the Preface for more information.



# 2 *Setup and Installation*

---

This chapter describes how to set up UniSite and get it working with your equipment. Before you read this chapter, make sure you have read the previous chapter, "Introduction."

This chapter guides you through configuring the hardware and installing the system software. The process is divided as follows:

1. Install a Module ..... 2-3
2. Check the Line Voltage ..... 2-5
3. Choose Your Configuration ..... 2-5
4. Connect Your Equipment ..... 2-6
5. Insert the System Disk ..... 2-17
6. Install the Base ..... 2-20
7. Power Up the Programmer ..... 2-21
8. The Power-on Screen ..... 2-26
9. Insert the Algorithm Disk(s) ..... 2-28

The following steps are optional, depending on your programmer configuration.

10. Update System Software ..... 2-28
11. Set Up High Speed Download ..... 2-29
12. Install and Update the MSM ..... 2-32

When you have finished with steps 1 through 9 and any optional steps 10 through 12, do the following:

- Refer to step 13, "Install Devices," on page 2-34 for instructions on how to install devices.
- Refer to the "What to Do Next Time" section on page 2-43 for instructions on what to do next time you powerup your programmer.

The following maintenance information is also included:

- Changing the Line Voltage ..... 2-40  
Replacing the Line Fuse ..... 2-40

## Before You Begin

Before you begin the setup and installation, make sure you have read and understand the terms of the Software License Agreement, which is printed on the outside of the envelope containing the Algorithm/System disk(s).

### Licensing Agreement

**IMPORTANT:** If you do not agree with the terms of the licensing agreement, do not open the software package. Opening the package indicates that you have accepted the terms of the agreement.

### Definitions

Some terms with which you might not be familiar are described below.

#### Algorithm Disks

Algorithm disks are inserted into the **programmer's** disk drive and contain the system software and all the programming algorithms for the devices currently supported by the programmer. Your programmer package includes at least one Algorithm disk (the number of disks your programmer package contains can be found on the disk labels). Unless you have the RAM Device Selection feature enabled or have your algorithm files stored on a MSM (mass storage module), one of the Algorithm disks must be installed each time you select a device (the programmer should prompt you for which disk to insert).

#### Device List Disk

The Device List disk is inserted into a **DOS-based PC's** disk drive and contains a detailed list of device support and information. In addition, the Device List program enables you to generate custom device lists (you can locate a list of the specific devices you use and optionally print that list or save it to file). The data in the Device List is formatted to make importing it into a database or spreadsheet easy.

For more information, see the **readme.txt** file, which is located on the Device List disk.

#### Host

A minicomputer, such as a Sun, DEC, or Apollo workstation. You can use the workstation to control the programmer and for remote storage of data files.

#### PC

A DOS-based personal computer, such as an IBM PC or compatible, or other microcomputer. The PC can be used to control the programmer and can be used for remote storage of data files.

#### Programmer

A generic name for UniSite and other device programmers.

#### System Disk

The System disk is inserted into the **programmer's** disk drive and contains the system software and the configuration files used when you boot the programmer.

#### Terminal

A stand-alone terminal, such as the DEC VT-100, Qume VT-101, and the Wyse WY-30/40/70 family of terminals.

#### Utility Disk

The Utility disk is inserted into a **DOS-based PC's** disk drive and contains HiTerm software for your PC. For more information, see the **manual.doc** file, which is located on the Utility disk.

---

## 1. Install a Module

This section tells you how to insert a module into UniSite. Although these might seem like a simple procedure, we strongly recommend you read this section.

### About Modules

A module serves as the interface between the device and the programmer. The modules available at the time of the printing of this manual are listed in the "Options" section of Chapter 1. For a current list of all available modules, or for more information, contact Data I/O Customer Support as listed in the Preface.

### Removing the Protective Cover

When you receive UniSite, a protective cover might be installed in place of a large module. To remove the protective cover, snap it off.

If you choose, you can leave the protective cover in place until you want to install a large module, such as PinSite or SetSite.

### Installing a Module

UniSite is designed to accept two modules: a small one on the left and a large one on the right. Each module allows you to support a specific package style, or support special types of programming operations such as set or gang programming.

Follow the instructions below to install a small module and a large module.

### Installing a Small Module

To install a small module, such as Site48HS, follow the steps below:

---

**CAUTION:** *Remove any devices in a module before you install the module in UniSite.*

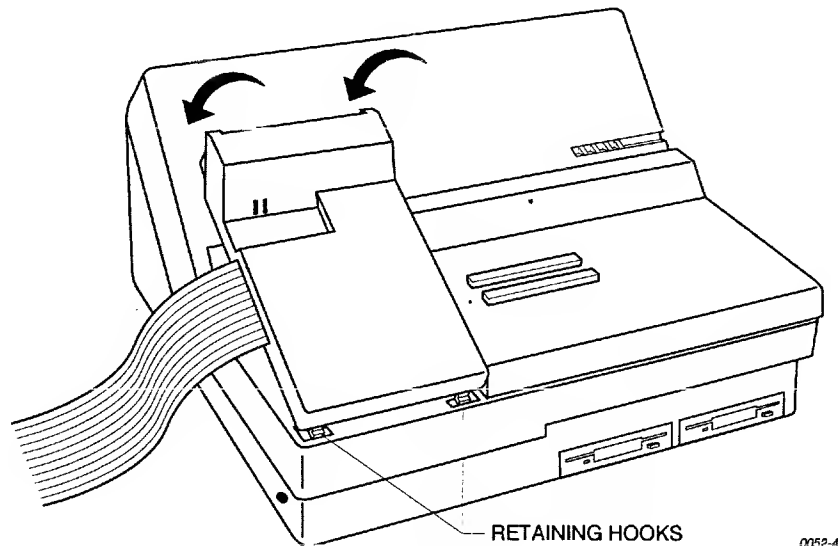
1. Insert the two retaining hooks on the bottom of the small module into the two slots on the top left side of UniSite. Figure 2-1 shows the location of the retaining hooks on the module and also shows the location of the slots for the small module on UniSite.
2. Carefully lower the back of the small module until the module connector touches its mating connector on UniSite.
3. To ensure complete contact, firmly press down on the rear of the small module.

You do not need to use excessive force when pressing on the module.

---

**CAUTION:** *You can damage UniSite by exerting too much force on the module.*

**Figure 2-1**  
*Installing a Module*



### Installing a Large Module

To install a large module, follow the steps below:

1. Insert the two retaining hooks on the bottom of the large module into the two slots on the top right side of UniSite. Figure 2-1 shows a large module already inserted into UniSite.
2. Carefully lower the back of the large module until the module connector touches its mating connector on UniSite.
3. To ensure complete contact, firmly press down on the rear of the large module.

You do not need to use excessive force when pressing on the module.

---

**CAUTION:** *You can damage UniSite by exerting too much force on the module.*

### Removing a Module

To remove a large module or a small module, follow the steps below:

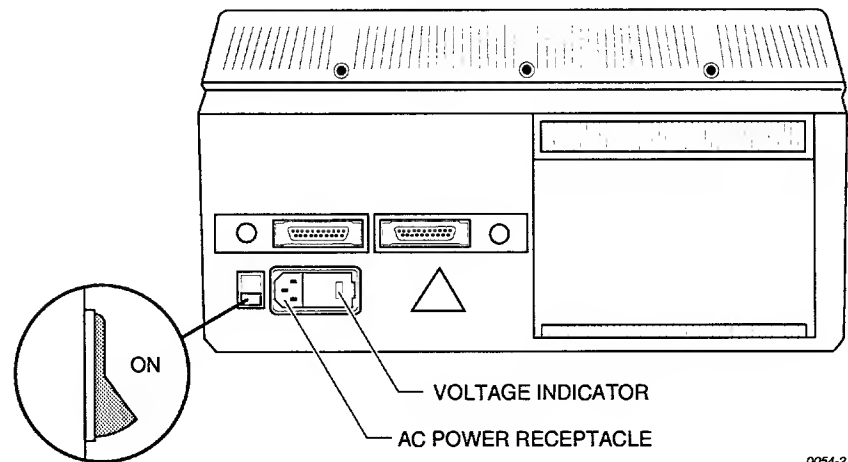
1. Make sure you are not performing any device operations, such as loading, programming, or verifying.
2. Make sure the Active LED on the module is not lit.
3. Remove any devices in the module.
4. Lift the rear of the module until the retaining hooks on the front of the module are disengaged from the slots on the top of UniSite.
5. Set the module aside.

---

## 2. Check the Line Voltage

Verify that the line voltage is correct by checking the line voltage indicator on the rear panel. Figure 2-2 shows the location of the voltage indicator.

**Figure 2-2**  
*UniSite's Rear Panel*



0054-2

---

**CAUTION:** *Damage to the equipment may occur if the instrument is operated with the wrong voltage.*

If the line voltage is correct, continue with the next section.

If the line voltage is not correct, skip to the section titled "Changing the Line Voltage," which is located at the end of this chapter. Continue with the next section after you have changed the line voltage.

---

## 3. Choose Your Configuration

Review the equipment you have available and then decide which of the following configurations you will use to control UniSite.

- Connecting to a PC
- Connecting to a Host
- Connecting to a Terminal

Once you have decided the configuration you will use, skip to that section. For example, if you decide to connect to a terminal, skip to the section titled "Connecting to a Terminal."



## 4. Connect Your Equipment

This section is divided into three subsections:

- Connecting to a PC
- Connecting to a Host
- Connecting to a Terminal

Skip to the section that applies to your configuration.

### Connecting to a PC

Connecting UniSite to a DOS-based PC allows you to take advantage of all UniSite's capabilities. For this configuration, you must use terminal emulation software, which allows UniSite and the PC to communicate. Or, you can use TaskLink to control UniSite. If you are using TaskLink, see the *TaskLink Getting Started Guide* for setup information.

#### What You Need

To connect UniSite to a PC, you need the following:

- An unused RS-232C serial port on the PC. Usually serial ports on a PC are labeled COM1, COM2, etc.
- Terminal emulation software. We suggest you use HiTerm™, which is supplied with UniSite and enables you to download files to UniSite at up to 115.2K baud.
- A 25-pin serial cable. If you need more information about which type of cable will work with UniSite, or if you need to build your own cable, see the section later in this chapter titled "More About Cables."

#### Making the Connections

To connect UniSite to a DOS-based PC, follow the steps below.

1. Connect one end of the RS-232C serial cable to the serial port connector on the back of the PC.

While PC serial ports are usually labeled COM1, COM2, etc., some PCs may label serial ports differently. Consult the documentation that came with the PC for more information.

---

**CAUTION:** *To minimize electromagnetic interference, use only properly shielded and terminated cables.*

2. Connect the other end of the serial cable to the Terminal port on the back of UniSite.

## Installing HiTerm

HiTerm is provided on the Utility disk included with UniSite. You can run HiTerm from a floppy disk or hard disk. These instructions tell you how to install HiTerm on a hard disk. If you want to run HiTerm from a floppy disk, consult the *HiTerm User Manual*.

These instructions assume you have a working knowledge of DOS. If you have questions, consult your DOS manual or the HiTerm documentation.

## Why HiTerm?

As mentioned above, you must use terminal emulation software to control UniSite and to permit the transfer of data files between the PC and UniSite. Written especially for UniSite, HiTerm is a VT-100 terminal emulator that allows you to download files to UniSite at up to 115.2K baud. HiTerm automatically handles all the file management tasks, which makes sending and receiving files easier.

Follow the steps below to install HiTerm on your hard disk drive.

1. Copy all the files from the HiTerm disk to a directory on the PC's hard disk. For example, `C:\hiterm`.
2. Make sure the PATH statement in the `autoexec.bat` file includes the directory where the HiTerm files reside.
3. Edit the `prg9600.cfg` file so it reflects the setup of your PC. This configuration file, specifying the operating mode, communication parameters, and PC type, is read when HiTerm is invoked.

If HiTerm cannot read the configuration file, HiTerm will use the following default settings:

Mode: Programmer  
Baud rate: 9600  
Parity: None  
Data bits: 8  
Stop bits: 1  
COM port: 1  
PC type: Auto detect

---

*Note: The PC type parameter applies only to HiTerm version 3.10 and later. If you received an earlier version of HiTerm (3.01 for example) set the configuration file as described above but do not include the PC type parameter.*

When you edit the **prg9600.cfg** configuration file, be sure each line conforms to the specifications below.

- First line – mode. Enter either General (G) or Programmer (P). Specify Programmer mode for use with UniSite. Only the first character of the line is significant.
  - Second line – baud rate. The complete number is required (for example, 9600, not 96). See the *HiTerm User Manual* for a list of baud rates supported by HiTerm.
  - Third line – parity. Specify None, Odd, or Even (N,O,E). Only the first character of the line is significant.
  - Fourth line – data bits. Specify 7 or 8.
  - Fifth line – stop bits. Specify 1 or 2.
  - Sixth line – COM port. Specify 1 or 2.
  - Seventh line – PC Type. Specify IBM, NEC or Autodetect (I,N,A). Select N for NEC's PC-9800 family of PCs, I for IBM-compatible PCs, or A for Autodetect if you are not sure.
4. Edit the **program.bat** file to reflect the location of the configuration files. Edit the two lines that invoke HiTerm so they point to the drive and subdirectory containing the HiTerm files. An example of the **program.bat** file modified to reflect HiTerm's installation in the **c:\util\hiterm** directory is shown below.

---

*Note: In the following example, bold print indicates items you should modify if your configuration is different.*

```
echo off
Rem: HITERM will use the configuration filename
Rem: from command line if present.
If not (%1) == () HITERM c:\util\hiterm\%1

Rem: HITERM will use PRG9600.CFG if no
Rem: configuration file is specified.
If (%1) == () HITERM c:\util\hiterm\prg9600.cfg
```

5. Installation of HiTerm is now complete. Reboot your PC.

## The Next Step

UniSite is now connected to your PC. Skip to the section titled "5. Insert the System Disk" to continue with the setup and installation of UniSite.

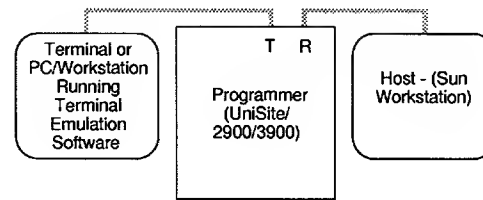
## Connecting to a Host

Connecting the programmer to a host allows you to use the host for remote file storage.

### Transparent Mode

Transparent mode, shown in Figure 2-3, is a feature of UniSite that allows the programmer to be inline between your host computer and a terminal. This eliminates the need for a switch box or a second link to the host and enables you to communicate with the host and download directly from the host to the programmer. The host could be a networked file server such as a VAX or a Sun. When set up properly, the terminal connected to the programmer can control both the programmer and the remote host.

Figure 2-3  
Transparent Mode



In Transparent mode, the programmer passes all characters through its Terminal and Remote ports as if it weren't there. The two serial ports on the programmer can even operate at different baud rates. While operating the programmer from the terminal, press **Esc Ctrl + T** to toggle the programmer between terminal mode and transparent mode. The programmer remains in transparent mode until it receives another **Esc Ctrl + T** command, at which time it switches back to terminal mode.

### What You Need

To connect the programmer to a host, you need the following:

- An unused RS-232C serial port on the host.
- A 25-pin serial cable. If you need more information about which type of cable will work with the programmer, or if you need to build your own cable, see the section later in this chapter titled "More About Cables."

### Making the Connections

To connect the programmer to a host, follow the steps below.

1. Connect one end of the RS-232C serial cable to the serial port connector on the host.

On Sun workstations the serial ports are usually labeled Serial Port A, Serial Port B, etc. Other brands of workstations may label serial ports differently. Consult the documentation that came with the workstation for more information.

---

**CAUTION:** *To minimize electromagnetic interference, use only properly shielded and terminated cable.*

2. Connect the other end of the serial cable to the Remote port on the back of the programmer.

### Checking the Communication Parameters

With the programmer connected to the host, adjust the communication parameters of the serial port connected to the programmer as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit
- Full duplex
- CTS/DTR handshaking

---

*Note: CTS/DTR (Hardware Handshake) is enabled as the default; however, if those signals aren't connected, the programmer will sense this and still communicate properly using XON/XOFF (Software Handshake). The programmer always uses XON/XOFF regardless of whether CTS/DTR handshake is enabled.*

Later, once you have established communication and the programmer is operating, you can change the communication parameters to suit your needs. Consult the operator's manual supplied with the host if you need to change the host's communication parameters.

### The Next Step

The programmer is now connected to your host. Skip to the section titled "5. Insert the System Disk" to continue with the setup and installation of the programmer.

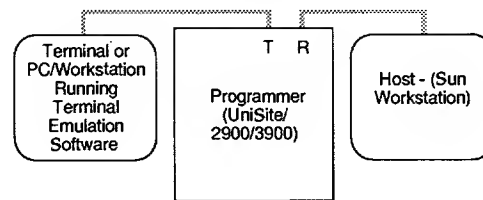
## Connecting to a Terminal

Connecting the programmer to a terminal is the simplest configuration. By connecting the programmer to a terminal, you can take advantage of all the programmer's capabilities.

## Transparent Mode

Transparent mode, shown in Figure 2-4, is a feature of the programmer that allows the programmer to be inline between your terminal and host computer. This eliminates the need for a switch box or a second link to the host, and enables you to download directly from the host to the programmer. The host could be a networked file server such as a VAX or a Sun. When set up properly, the terminal connected to the programmer can control both the programmer and the remote host.

**Figure 2-4**  
*Transparent Mode*



0953-2

In transparent mode, the programmer passes all characters through its Terminal and Remote ports as if it weren't there. The two serial ports on the programmer can even operate at different baud rates. While operating the programmer from the terminal, press **Esc** **Ctrl** + **T** to toggle the programmer between terminal mode and transparent mode. The programmer remains in transparent mode until it receives another **Esc** **Ctrl** + **T** command, at which time it switches back to terminal mode.

## What You Need

To connect the programmer to a terminal, you need the following:

- An unused RS-232C serial port on the terminal.
- 25-pin serial cable. If you need more information about which type of cable will work with the programmer, or if you need to build your own cable, see the section later in this chapter titled "More About Cables."

If you are going to operate the programmer in transparent mode, you will also need the following:

- An unused RS-232C serial port on the host. The host can be a VAX, a Sun, etc.
- 25-pin serial cable.

**Approved Terminals**

Before you connect your terminal to the programmer, make sure your terminal is compatible with the programmer. If your terminal type is listed below, then your terminal is compatible with the programmer.

- ANSI 3.64 compatible terminals
- DEC VT-100 compatible terminals
- Qume QVT-101 compatible terminals
- TELEVIDEO TVI-910 compatible terminals
- Wyse WY-30 compatible terminals

If your terminal is not included in the above list, check the documentation that came with your terminal to see if the terminal can emulate one of the terminal types listed above. If your terminal has programmable function keys, the following table lists the expected codes for the four function keys:

VT-100 Key	Expected Code	Wyse-30 Key	Expected Code
PF1	ESC O P	F1	SOH @ CR
PF2	ESC O Q	F2	SOH A CR
PF3	ESC O R	F3	SOH B CR
PF4	ESC O S	F4	SOH C CR

**Making the Connections**

To connect the programmer to a terminal, follow the steps below.

1. Connect one end of an RS-232C serial cable to the serial port connector on the back of the terminal.

On some terminals the serial port might be labeled Modem; on others the serial port might be labeled EIA. Consult the documentation that came with the terminal for more information.

---

**CAUTION:** *To minimize electromagnetic interference, use only properly shielded and terminated cables.*

2. Connect the other end of the cable to the Terminal port on the back of the programmer.

If you will not be using transparent mode, skip to the section titled "Checking the Communication Parameters." Follow the steps below if you will be using transparent mode.

3. Connect one end of an RS-232C serial cable to the serial port connector on the host. If the host is not available locally, i.e., the host is a networked VAX, connect the serial cable to the appropriate serial port.

---

**CAUTION:** *To minimize electromagnetic interference, use only properly shielded and terminated cables.*

4. Connect the other end of the cable to the Remote port on the back of the programmer.

**Checking the  
Communication  
Parameters**

Adjust the communication parameters of the equipment connected to the programmer as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit
- Full duplex
- CTS/DTR handshaking

---

*Note: CTS/DTR (Hardware Handshake) is enabled as the default; however, if those signals aren't connected, the programmer will sense this and still communicate properly using XON/XOFF (Software Handshake). The programmer always uses XON/XOFF regardless of whether CTS/DTR handshake is enabled.*

Once you have established communication and the programmer is operating, you can change the communication parameters to suit your needs. Consult the manual supplied with the terminal if you need to change the terminal's communication parameters.

If you are using transparent mode, set the communication parameters on the serial port on the host as described above. After you have powered up the programmer, you can change the parameters on the programmer's Remote port to match the communication parameters of the host.

The programmer is now connected to your terminal. Skip to the section titled "5. Insert the System Disk."

**More About Cables**

You will need one 25-pin RS-232C serial cable for each piece of equipment you will connect to the programmer.

Normally, when you connect equipment to a programmer, you must match Data Terminal Equipment (DTE) to Data Communications Equipment (DCE).

The programmer is compatible with both types of equipment and automatically configures the Terminal and Remote ports to be compatible with the equipment connected to it.

**SmartPort**

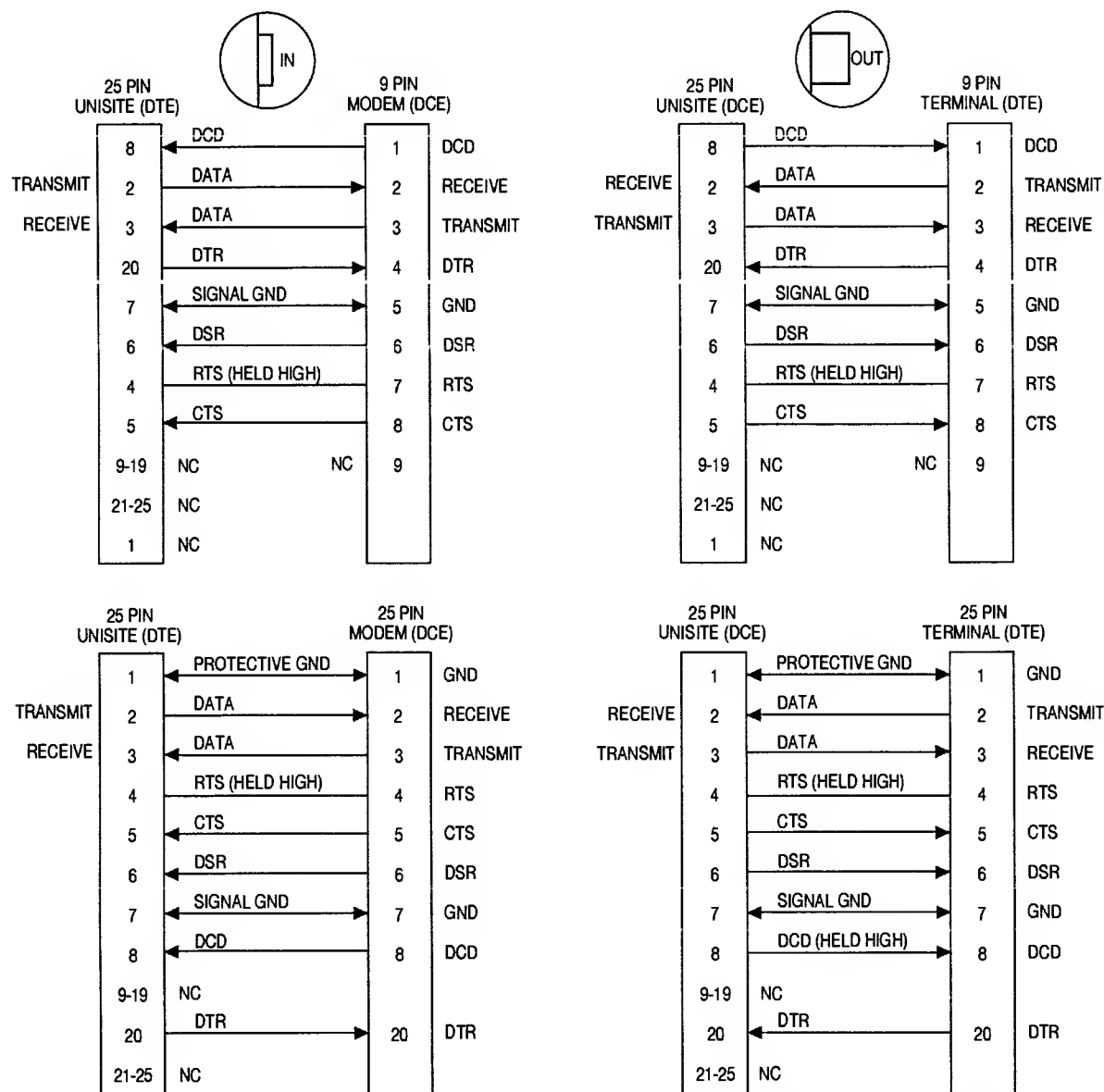
If you are not sure what type of equipment you have, don't worry. The programmer's SmartPort feature will automatically toggle between the two types until a connection is established. For now, just connect the cables. If you don't have 25-pin serial cables, or if you are not sure if they will work, you can use the cabling diagrams below to build your own cable.



## Making Your Own Cable

The programmer receives commands and sends responses through an RS-232C port using a 25-pin D connector in two possible configurations: either DTE or DCE. The connections are shown in Figure 2-5.

**Figure 2-5**  
Pin Designations for RS-232C  
Serial Port Connection



The minimum hookup includes Pins 2, 3, and 7.  
Pins 1 and 7 are tied together.

0362-4

## Switching Modes

As shown in Figure 2-5, the switches on the programmer's back panel toggle each port between DTE and DCE. For now, do not change the settings of the switches. For information when to use the DTE/DCE switches, see the section titled "5. Insert the System Disk."

**Pin Functions When In DTE Mode**

The following table explains the function of the pins on the Terminal and Remote ports when connected to DCE equipment.

Pin	Function	Description
1	Ground	Provides a safety ground connection.
2	Transmit Data	Carries the transmitted data.
3	Receive Data	Carries the received data.
4	Request to Send	This line is held high by the programmer.
5*	Clear to Send	A high on this line enables the programmer to transmit data. (Used for hardware handshaking.) A low inhibits data transmission from the programmer.
6*	Data Set Ready	This line is held high when the remote source is ready to send or receive data. A low inhibits data transmission from the programmer.
7	Signal Ground	Provides a reference ground for all signals on the cable.
8*	Data Carrier Detect	This line is held high when the modem detects a carrier. A low on this line inhibits the programmer from transmitting data.
9-19	No Connection	
20	Data Terminal Ready	This line is pulled high by the programmer to indicate it is ready to receive data. This line is pulled low to signal the remote computer to stop sending data. (Used for hardware handshaking.)
21-25	No Connection	

\* If these lines are not connected, the programmer will consider them high and will function normally.

**Pin Functions When In DCE Mode**

The following table explains the function of the pins on the Terminal and Remote ports when connected to DTE equipment.

Pin	Signal	Description
1	Ground	Provides a safety ground connection.
2	Receive Data	Carries the received data from the DTE device to the programmer.
3	Transmit Data	Carries the transmitted data from the programmer to the DTE device.
4	Request to Send	This line is held high by the programmer.
5	Clear to Send	A high on this line from the programmer means that it is ready to receive data. (Used for hardware handshaking.)
6	Data Set Ready	This line is held high when the programmer is ready to transfer data.
7	Signal Ground	Provides a reference ground for all signals on the cable.
8	Data Carrier	This line is held high by the programmer.
9-19	No Connection	
20*	Data Terminal	A high on this line enables the programmer to transmit data. (Used for hardware handshaking.) A low inhibits data transmission from the programmer.
21-25	No Connection	

\* If this line is not connected, the programmer will consider it high and will function normally.

## 5. Insert the System Disk

Follow the steps below to insert the System disk into UniSite.

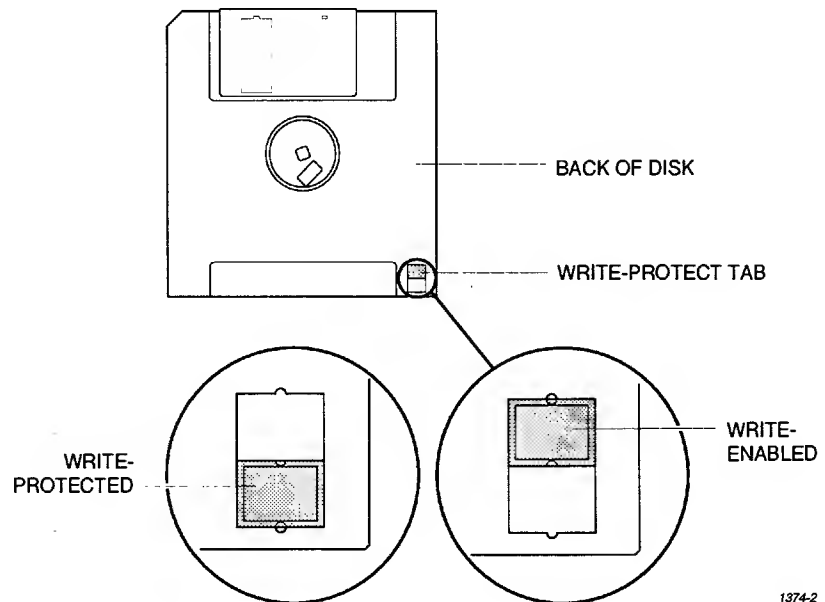
1. Your UniSite software package contains the programmer disks. The first time you open the software package, remove the programmer disks from the software envelope. (You should receive several disks: the System disk, Algorithm disks, Device List disk, and Utility disk.)

---

*Note: Fill out and return the End User Registration Card located in this manual. After registering with Data I/O, you will receive information about all future updates and product upgrades.*

2. Make sure both the System disk and the Algorithm disks are write enabled. Figure 2-6 illustrates how to write-enable a disk.

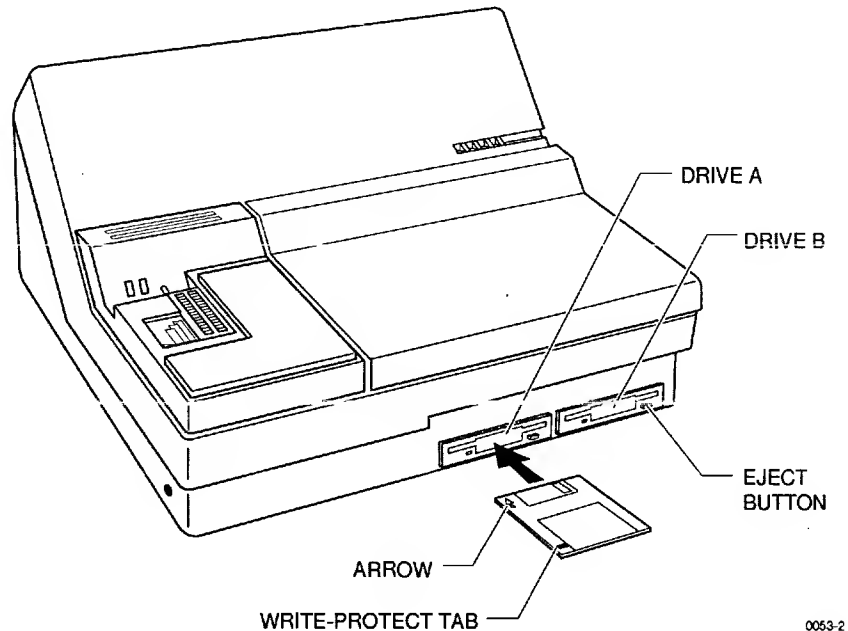
**Figure 2-6**  
Write Enabling a 3.5-inch Disk



1374-2

3. As shown in Figure 2-7 insert the System disk into drive A, making sure the arrow molded into the plastic case is on the top of the disk and is pointing toward UniSite. Push the disk straight into the drive until the disk drops down and the eject button pops out.

**Figure 2-7**  
*Inserting the System Disk*



4. If you have a two-disk drive UniSite, you can insert an Algorithm disk into drive B (the drive on the right). For more information, see "9. Insert Algorithm Disk," on page 2-28.

---

*Note: Do not attempt to use the System disk or an Algorithm disk with more than one UniSite. Each disk is configured so that it can only be installed on one programmer. Once you get the programmer set up, make a backup copy of the System disk and the Algorithm disk(s).*

## Software Version Compatibility

The UniSite System and Algorithm disks contain separate version numbers for their contents. UniSite checks the version numbers of the disks and informs you if they are not compatible with each other. The version numbers of the disks do not have to be identical for the disks to work together.

A System disk and the Algorithm disks are compatible if they meet the following rules:

- All digits to the left of the decimal point are the same.
- The first digit to the right of the decimal point is the same. (Any additional digits, if present, do not have to match.)

For example:

<u>System Version</u>	<u>Algorithm Version</u>	<u>Compatible?</u>
2.50	2.50	Yes
2.51	2.50	Yes
2.50	2.51	Yes
2.50	2.40	No

This version control system allows the Algorithm disks to be updated without requiring a new System disk.

## 6. Install the Base

A base is installed in a module that supports the base (not all modules require a base). The Base serves as the interface between the device and the programmer. Some Bases (such as the PLCC Base) are designed to be used with MatchBook device carriers. The MatchBook device carrier holds a device in place on the Base (MatchBooks are described on page 2-34). Some bases (such as the PPI Base) use device adapters.

Follow the instructions in this section if the module installed on UniSite requires a base. For additional information, refer to the documentation included with the base and refer to "Installing Devices" on page 2-34.

### Installing a Base

Follow the procedure below to install a Base in the programmer.

---

*Note: You can install and remove a Base with the power on as long as you are not performing a device operation.*

1. Position the programmer so that the front points toward you.
2. Examine the module opening for the base. It has a sliding handle on the front half and notches on the back half. Pull the sliding handle toward the front.

---

**CAUTION:** *Poking a foreign object into the Base opening will damage the programmer.*

3. Align the notches on the Base with the notches in the opening.
4. Insert the Base into the opening, making sure the guide pins in the Base opening line up with the holes in the Base.
5. Finally, squeeze the handles together, locking the Base in place. Do not use excessive force when compressing the handles.

---

**CAUTION:** *You can damage the programmer by squeezing too hard on the handles.*

### Removing a Base

When removing a Base from the programmer, be sure to apply even pressure while moving the handles apart. If you exert uneven pressure on the handles, you could damage the sliding handle.

To remove a Base, follow the steps described below.

1. To remove the Base, separate the handles with your thumbs and fingers. Push back on the Base handle with both thumbs while pushing forward on the sliding handle with your forefingers.
2. Lift the Base up and out of the module. Store the Base in a safe place.

---

**CAUTION:** *You must apply even pressure while moving the handles apart. If you exert uneven pressure on the handles, they may jam in the tracks. Apply an even force to realign the handles.*

---

## 7. Power Up the Programmer

To power up the programmer, follow the steps below.

1. Make sure the programmer is positioned so the fan on the bottom will not be obstructed.
2. If you will be using an antistatic wrist strap, put on the wrist strap and connect it to the programmer. The ground connector for the antistatic wrist strap is on the back panel of the programmer. Refer to Figure 1-3 for the location of the ground connector on the programmer.

---

**WARNING:** To help prevent electric shock, the antistatic wrist strap must contain a  $1\text{M}\Omega$  (minimum) to  $10\text{M}\Omega$  (maximum) isolating resistor.

### Connecting the Power Cord

Connect one end of the ac line cord to the ac receptacle on the rear panel of the programmer and the other end to a properly grounded ac outlet.

---

**WARNING:** To ensure proper grounding, and to avoid hazard of electrical shock, connect the programmer **ONLY** to a properly grounded ac outlet.

The programmer contains a switching power supply that configures itself to operate on the proper voltage. The power supply accepts voltages ranging from 90 to 264 Vac and frequencies ranging from 48 to 63 Hz.

Make sure that a module is installed in the programmer. Also, make sure the device socket in the Base is empty.

---

**CAUTION:** *Leaving a device in the socket during powerup will cause power-up self-test failures and could damage the device.*

Power up the PC, workstation, or terminal. If you will be controlling the programmer from a PC or workstation, make sure that the terminal emulation software (such as HiTerm) is running. If you will be controlling the programmer from a terminal, make sure that the terminal is in the proper emulation mode (such as VT-100 mode). See the "Approved Terminals" section earlier in this chapter for more information.



## Powering Up

Power up the programmer. The power switch is located on the back panel. When you turn the power switch on, the Power LED should light. If it doesn't, turn the programmer off, check the power connections, and turn the programmer on again.

---

*Note: Do not remove the **System** disk while either the Self Test or disk drive LED is lit. If you remove the System disk during powerup, you will need to reboot the programmer.*

### Did the Programmer Pass Self-test?

The programmer always performs a self-test upon powerup. The four front panel LEDs illuminate in different patterns, depending on the results of the self-test. The programmer has completed powerup when the Self Test LED and disk drive LED are off. If the Self Test LED does go off, then go to the section titled "Are the Right LEDs Lit?"

If the powerup self-test detected anything wrong, the four LEDs illuminate in different patterns, telling you what the self-test found. In general, if one or more front panel indicators is blinking after the self-test, there may be a faulty circuit board in the programmer. Contact your nearest Data I/O Customer Support office for more information. The different combinations of blinking LEDs are explained below.

----- Indicator -----				
Power	Terminal	Remote	Self-Test	Description
On	Off	Off	On	Self-test in progress.
Off	X	X	On	Bad power supply
On	Blinking	Blinking	Blinking	Power Supply problem; check voltage selector.
On	Blinking	Off	On	Bad CPU or EPROM
On	Off	Blinking	On	Bad system RAM
On	Blinking	Blinking	On	Bad serial port DUART
On	On	Off	Off	Terminal port connected; self-test finished. No errors.
On	Off	On	Off	Remote port connected; self-test finished. No errors.
On	On	On	Off	Remote & terminal ports connected; self-test finished. No errors.

---

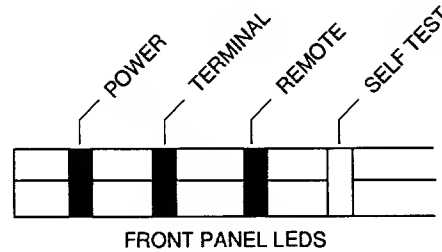
*Note: X = don't care condition.*

*"Terminal port OK" means that the programmer has detected an RS-232 serial device connected to the Terminal port and that a valid connection has been established with that device.*

**Are the Right LEDs Lit?**

If the programmer completed the self-test successfully, the Power LED will be lit. Also, if you have equipment connected to the Terminal and Remote ports, the corresponding LEDs should be lit. Figure 2-8 shows the front panel indicators.

**Figure 2-8**  
*Front Panel Indicators*



0361-2

If all the right LEDs are lit, go to the section titled "Is the Power-on Screen Displayed?"

If the Remote LED and/or Terminal LED should be lit and they are not, check the connections between the programmer and the connected equipment. If all connections appear correct, refer to "Configuring the Ports" below.

**Configuring the Ports**

Before communication between UniSite and a device connected to one of UniSite's serial ports can be established, three requirements must be met:

- UniSite's port must be configured using the DCE/DTE button so that it is compatible with the equipment it is connected to.
- UniSite's serial I/O parameters must match those of the equipment it is connected to.
- UniSite must recognize the type of terminal that you are using. (Selecting the correct terminal type is described in the next section, "8. The Power-On Screen.")

The steps below tell you how to configure the Terminal port. Use the same procedure for configuring the Remote port.

1. After you have powered up UniSite and the Self-Test indicator has gone out, look at the Terminal LED on the front panel.
2. If the LED is lit, UniSite's Terminal port is configured correctly. Skip to the section titled "Is the Power-on Screen Displayed?"
3. If the LED is not lit, press the button next to the Terminal port on the rear panel of UniSite once. (Remember, if you are configuring the Remote port, press the button next to the Remote port.) When the switch is pressed in, UniSite acts as Data Terminal Equipment (DTE); when in the out position UniSite acts as Data Communications Equipment (DCE).

After you press the button, the LED should light. If it does light, skip to the section titled "Is the Power-On Screen Displayed?"

If the LED does not light, continue with the section titled "Checking the Connections."

### Checking the Connections

Sometimes problems are caused by unconnected cables. Turn the programmer off and check all of the following:

- Power cords – Are they all plugged into a live outlet and into the equipment?
- Cables – Is each cable between the programmer and a peripheral connected properly? Is each cable connected to the proper port?
- Terminal – Is the terminal plugged in and turned on? Are the display controls adjusted to allow viewing? Is the terminal an approved terminal type? (See the list of approved terminals for more information) Are the communication parameters set correctly? Is the cable connected to the proper port?
- Host – Is it plugged in and turned on? Is the terminal emulation software installed and configured properly? Is the terminal emulation software running properly? Are the display controls on the monitor adjusted to allow viewing?
- Disk – Is the System disk inserted properly?
- Module – Is there a Module inserted into the programmer? Is it inserted properly?
- Base – Is there a Base inserted into the Module? Is it inserted properly? (A Base should be installed if the Module requires one.)
- Devices – Is the device socket(s) empty? (Sockets must be empty for the programmer to boot up properly.)

Now, after checking everything described above, reboot the programmer and go to the beginning of this section, "Power Up the Programmer."

### Rebooting the Programmer

If you need to reboot the programmer, you can do it in either one of the following two ways:

- Turn the programmer off, wait a few seconds, and then turn the programmer on again.
- Press **Esc** **Ctrl** + **W** .

The programmer always performs a powerup self-test whether you do a cold boot (cycling power) or a warm boot (pressing **Esc** **Ctrl** + **W** ).

### Is the Power-on Screen Displayed?

If the Power-on screen (shown in Figure 2-9) is displayed, you have successfully booted up the programmer. Go to the next step, "8. The Power-on Screen."

If you do not see the Power-on screen, or if you see random characters, the programmer is not communicating properly with your controlling terminal/workstation. One possible cause of the random characters is that the baud rates of the programmer and the controlling equipment do not match. In this case, follow the procedure in the next section to execute the AutoBaud function to "sync up" the baud rates.

**AutoBaud and Baud Rates**

When enabled, AutoBaud determines the baud rate of the equipment connected to the programmer's Terminal port and sets the programmer's baud rate to match.

---

*Note: AutoBaud is available only from the Power-on screen and works only on the port configured as the User Menu Port. The User Menu Port is the port the programmer sends the user interface data, such as screens and online help, through. When shipped from the factory, the Terminal port is configured as the User Menu Port. So, for "out of the box" setup, AutoBaud should work on the Terminal port.*

To execute AutoBaud, press **Break** and then **A** .

After executing AutoBaud, you should see the Power-on screen. If you see the Power-on screen, you have successfully booted up the programmer. Go to the next step, "8. The Power-on Screen."

If you do not see the Power-on screen, you should go back to the section titled "Checking the Connections."

**When to Use AutoBaud?**

Normally you can set the communication parameters on the controlling PC/workstation/terminal to match the programmer's baud rate. Use AutoBaud when you will not be able to set the controlling equipment's baud rate to match the programmer's. For instance, use AutoBaud to control the programmer with a terminal that is not capable of operating at 9600 baud. In this example, you would set the terminal's baud rate as close to 9600 as possible and then execute AutoBaud when the programmer boots up.

## 8. The Power-on Screen

The Power-on screen, shown in Figure 2-9, appears after the programmer completes its powerup self-test successfully. The Power-on screen lists the version and configuration information for the programmer and the current and default terminal types.

Figure 2-9  
Typical Power On Screen

```

#####
***  ***  ***  ***  ***  #####  ***  #####  #####
***  ***  #####  ***  ***  **  **  ***  ***  ***
***  ***  #####  ***  ***  **  ***  ***  ***
***  ***  #####  ***  ***  **  ***  ***  ***
***  ***  ***  #####  ***  **  ***  ***  ***
***  ***  ***  ***  ***  **  ***  ***  ***
***  ***  ***  ***  ***  **  **  ***  ***  ***
#####  ***  ***  ***  #####  ***  ***  #####

DATA I/O UNIVERSAL PROGRAMMER
(Copyright 1987-1995 Data I/O Corp.)

SYSTEM CONFIGURATION:
User RAM: 8704 KB
PSM: Site40 V01      FSM: none
Mainframe pin drivers = 28 (4 pin drivers/board)
Software revision     = Y.YY
EPROM revision        = Z.Z
Algorithm revision     = X.XX
Current terminal type  = UYSE UY-30, TELEVIDEO TUI-910
Do you want to select a new terminal type? (Y/N) [N]:

```

*Note: The power-on screen your programmer displays might be slightly different from the screen shown above.*

At this point, you are asked if you want to choose a new terminal type. If the current terminal type is correct, press ☐ and go to the section titled "Setting Up High Speed Download."

To change the current terminal type and/or the default terminal type, refer to the appropriate section below, "Selecting a New Terminal Type" or "Changing the Default Terminal." If you will be using the current configuration for a while, we suggest you change the default terminal type to match the terminal type you will be using to control the programmer. For information on changing the default terminal type, go to the section titled "Changing the Default Terminal."

### Selecting a New Terminal Type

The following steps describe how to change the current terminal type. The change in terminal type will remain in effect until you reboot the programmer or change the terminal type.

1. At the bottom of the Power-on screen, the programmer displays the following prompt:

Do you want to select a new terminal type? (Y/N) [N]:

Press ☐ ☐ to select a new terminal type.

2. The programmer then displays the default and current terminal types and a list of the available terminal types. To select a new terminal type, enter the number corresponding to that terminal type and press ☐.

---

*Note: If the screen is blank or if random characters appear after you press Enter, the Terminal type configuration is incorrect. Switch the power off and then power up the programmer. Then return to the beginning of this section and select the proper terminal type.*

If you do not see your terminal listed on the screen and do not know what terminal type(s) it can emulate, refer to the Compatible Terminals list shown previously in this chapter.

After you have changed the terminal type for this current session, the programmer responds with the following prompt:

Save terminal type as power on default? (Y/N) [N]

3. Press **[N]** **[ ]** to go to the Main Menu without changing the default terminal type.

When the Main Menu appears, the programmer is ready for operation. Go to the section titled "9. Insert Algorithm Disk."

## Changing the Default Terminal

Follow the steps below to change the default terminal type.

1. At the bottom of the Power-on screen, the programmer displays the following prompt:  
Do you want to select a new terminal type? (Y/N) [N]:  
Press **[Y]** **[ ]** to select a new terminal type.
2. The programmer then displays the default and current terminal types and a list of the available terminal types. To select a new terminal type, enter the number corresponding to that terminal type and press **[ ]**.

---

*Note: If the screen is blank or if random characters appear after you press Enter, the Terminal type configuration is incorrect. Switch the power off and then power up the programmer. Then return to the beginning of this section and select the proper terminal type.*

If you do not see your terminal listed on the screen and do not know what terminal type(s) it can emulate, refer to the Approved Terminals list shown previously in this chapter.

After you have changed the terminal type for this current session, the programmer will respond with the following prompt:

Save terminal type as power on default? (Y/N) [N]

3. Press **[Y]** **[ ]** to change the default terminal type. After the programmer saves the currently selected terminal type as the power-on default, the Main Menu appears, indicating that the programmer is ready for operation. Go to the section titled "9. Insert Algorithm Disk."

---

## 9. Insert Algorithm Disk

Once UniSite has finished booting (the self test is finished and the drive LED is off), insert the Algorithm disk that contains the programming algorithms for the devices you plan on using into drive B (the drive on the right), or if your UniSite has only one disk drive, remove the System disk and insert the Algorithm disk. (If you are unsure which Algorithm disk to insert, insert the Algorithm disk that contains Algorithm Set 1.)

At various times, such as when you select a device, UniSite accesses the Algorithm disk. You might be prompted to insert a different Algorithm disk than the one currently in the programmer. If so, insert the disk for which you are prompted. For instance, if UniSite prompts you to Insert Algorithm Set 3 Disk, insert the Algorithm disk that contains Algorithm Set 3. If UniSite prompts to Insert Algorithm Disk, insert any one of the Algorithm disks.

Once you have inserted the Algorithm Disk, complete steps 10 through 13 as necessary. Refer to "What to Do Next" on page 2-43 for instructions on what to do the next time you power up your programmer.

---

## 10. Update System Software

When you have a new version of system software you want to use with UniSite, simply power up UniSite with the System disk containing the new system software. Your system software is automatically updated.

---

*Note: Instead of displaying the Main Menu after startup, the Programmer ID Screen is displayed. Press PF1 to return to the Main Menu. To view the programmer ID again, select the **Programmer ID** command as described on page 5-42.*

After you update your software, you might wish to use previous configuration files with the new software. To carry configuration files forward, refer to page 5-43.

### PROMlink-6 and TaskLink Users

If you are using PROMlink-6 or TaskLink, remember to select the **Update Device List** command from the **Utilities** menu in PROMlink-6 or TaskLink after updating your programmer to a new version of system software.

### When You Are Finished

When you are finished updating your UniSite, we suggest you make a backup copy of your new Algorithm and System disks. See the Duplicate Disk command on page 5-82 for more information on making a backup copy of your Algorithm and System disks.

---

*Note: The backup disks must first be formatted using the UniSite Format Disk Operation.*

## 11. Set Up High Speed Download

This section explains how to set up the programmer so you can download files to the programmer at 115.2K baud.

To take advantage of the High Speed Download feature, you must be controlling the programmer from a PC, the PC must be connected to the programmer's Remote port, and you must be using HiTerm as your terminal emulation software. See the section titled "Installing HiTerm" earlier in this chapter.

Skip to the section titled "What to Do Next Time" if you are not using HiTerm on a PC.

### Why Should I Use High Speed Download?

High Speed Download can reduce the transfer time for a large data file by as much as 92%.

As an example, downloading a formatted data file for a 1MB EPROM takes 6 minutes and 38 seconds at 9600 baud. With High Speed Download, it only takes 32 seconds to download the same data file from a 25 MHz 386-based PC.

---

*Note: To obtain the best results, we suggest you run HiTerm on at least a 286-based PC.*

### What Happens During High Speed Download?

When you begin a download to the programmer, the programmer and HiTerm communicate with each other to coordinate the transfer. HiTerm translates the data file on the PC into a special binary format. The compression of the data file reduces the size of the data file by up to 64%.

---

*Note: HiTerm compresses a temporary copy of the data file; your original data file is not changed.*

HiTerm then switches the baud rates on the PC and the programmer to 115.2K baud and downloads the data file to the programmer. After the download, the baud rates on the PC and programmer are restored to their original values.

### Configuring the Remote Port

For High Speed Download to work, you must have your PC connected to the programmer's Remote port and the programmer must be configured to run from the Remote port. Follow the steps below to configure the programmer to run in Terminal mode from the Remote port.

1. Start HiTerm. Power up the programmer. If the programmer is already powered up, reboot it by pressing **[Esc] [Ctrl] + [W]**. Complete booting up as normal.
2. From the Main Menu, press **[M] [C] [E] [S]** to get to the Serial Port Configuration screen.



3. The Mode: field for the Terminal port and the Remote port should match. If the two Mode: fields match, skip to step 4.

If the Mode: fields are different, toggle the DTE/DCE switch for the Remote port on UniSite's back panel. Figure 1-3 shows the location of the DTE/DCE switch for the Remote port. Go back to the beginning of this step after you have toggled the DTE/DCE switch on the Remote port.

4. From the Main Menu, press **[M][C][E][C]** to get to the Communication Parameters screen.
5. Move the cursor to the High Speed Download field. Press **[Y]** to enable High Speed Download. When you press **[Y]** UniSite sets the High Speed Download parameter to Y and displays the following message in the message bar:

Hit return to switch user menu port, ^Z to abort

UniSite also sets the User Menu Port parameter to R, which configures UniSite to send its user interface data to the Remote port.

6. Press **[J]**. You do not see any response on the screen.
7. Look at the back panel of UniSite. Remove the cable connecting UniSite and the PC from the Terminal port and connect the cable to the Remote port.

When you switch the cable from the Terminal port to the Remote port, the Terminal LED should go out and the Remote LED should light. If the Remote LED does not light, power down UniSite, reconnect the PC cable to the Terminal port, and go back to step 1.

8. After you have connected the cable to the Remote port, press **[Ctrl] + [R]** to redisplay the Communication Parameters screen. If you cannot redisplay the Communication Parameters screen, power down UniSite, reconnect the PC cable to the Terminal port, and go back to step 1.
9. Press **[F1]** to display the Main Menu. If the Main Menu appears, continue with the next step. If you do not see the Main Menu, power down UniSite, reconnect the PC cable to the Terminal port, and go back to step 1.

High Speed Download is now enabled.

#### About the User Menu Port

In the previous procedure, you configured UniSite for High Speed Download. As part of the configuration, you switched the User Menu Port from the Terminal port to the Remote port. The User Menu Port is the port through which the user interface data for UniSite is sent. User interface data includes screens, menu information, and online help. When shipped from the factory, the Terminal port is the User Menu Port.

## Changing the Powerup Defaults

The previous procedure told you how to change the User Menu Port for the current session. The following procedure tells you how to make that change part of your default working environment on the programmer.

If you do not save the changes you made in the previous procedure, you can still use High Speed Download for the current session. But, the next time you power up the programmer, you will have to repeat the procedure to enable High Speed Download.

### About Powerup Defaults

As part of the powerup process, the programmer reads a configuration file that contains the settings to use for over 50 system parameters. Collectively, the settings of these parameters are known as the Powerup Defaults. By comparison, the Powerup Defaults are similar to the `autoexec.bat` and `config.sys` files on a PC.

The parameters that comprise the Powerup parameters are listed and described in Chapter 5, "Commands."

Follow the steps below to make High Speed Download part of your powerup defaults.

1. From the Main Menu, press **[M][C][S]** to display the Save System Parameters screen. The programmer displays up to ten configuration files.

If you want to save High Speed Download as part of your powerup defaults, press **[1][J]**. The programmer displays the following message in the message bar:

Parameter Entered

2. Press **[J]** to save the current settings as the Powerup Defaults. The action symbol rotates while the programmer is saving the current parameter settings as the Powerup Defaults. When finished, the programmer displays the following message in the message bar:

System parameters saved

With High Speed Download saved as part of your Powerup Defaults, you will be able to use High Speed Download for the current session and for any future time you power up the programmer.

## Using High Speed Download

With the programmer properly configured for High Speed Download, you can use HiTerm to download files from the PC to the programmer at 115.2K baud.

For an example of downloading files to the programmer using High Speed Download, see Session 7 in Chapter 4.

---

## 12. Install and Update the MSM (Mass Storage Module)

The Mass Storage Module is an internal hard drive that can be installed in UniSite. The Mass Storage Module allows you to store system and algorithm files in the programmer, which can speed up the start-up and disk access routines.

### Hardware Installation Instructions

The installation instructions can be found in the Installation Guide that is shipped with the Mass Storage Module.

### Software Installation and Update

After installing the Mass Storage Module, you can set up your programmer to boot system software and read algorithm files directly from the MSM.

### Use Previous Configuration Files

**Before you install or update software on the MSM, if you wish to use previous configuration files located on the MSM, do the following to carry the configuration files forward:**

---

*Note: For more information about carrying configuration files forward, refer to page 5-43.*

1. Power up your programmer with your new System disk.
2. Copy the sysparm.sys file to the new System disk using the More Commands/File Operations/Copy File command.
  - a. In the Copy data file from field, enter  
H:\sysparm.sys
  - b. In the Copy data file to field, enter  
A:\sysparm.sys
  - c. Press **[Enter]** to copy the file. When UniSite displays the message Destination file already exists, press **[Enter]** to overwrite the file.

---

*Note: We suggest you work with the backup copies of your System and Algorithm disks and not with the originals.*

3. Remove the new System disk from UniSite.
4. In the More/Configure system/Restore screen, press **[F4]** to reconstruct the parameter list.

---

*Note: If you do not reconstruct the parameter list, the next time your programmer is powered up with the MSM, it prompts System parameter field is out of date. Do you want to update it (Y/N). Type Y and press **[Enter]** to reconstruct the parameter list.*

**Installing Software For The First Time**

To set up your programmer to boot from the MSM, install the system software on the MSM using the More Commands/Configure System/Mass Storage command (see page 5-55).

To set up your programmer to read algorithm files from the MSM, install the algorithm files on the MSM using the More Commands/Configure System/Mass Storage command (see page 5-55).

**Updating Software**

To update system and algorithm files located on the MSM, use the More/Configure/Mass Storage command (see page 5-55).

**Booting from the MSM**

After installing the system software on the MSM (as described above), you can boot directly from the MSM by doing the following:

1. Make sure that drive A is empty.
2. Reboot or powerup UniSite.

When UniSite boots up and does not find a disk in drive A, it boots using the system software located on your MSM.

**Storage Capacity**

The MSM is partitioned into 4 logical drives, with specifications as follows:

Drive	Storage Space	Maximum Number of Files	Data Type
C	31 MB	512	User data
D	31 MB	512	User data
H	7 MB	320	System data
I	10 MB	320	System data

**Storage Suggestions**

Drives C and D are reserved for user data and drives H and I are reserved for system use. Despite the fact that drives H and I can be written to and read from, we STRONGLY suggest you use only C and D to store your data.

**Limitations**

Other than the file operations listed below, drives C and D on the MSM can be used for all file operations that can be done with a floppy drive:

File Operation	Comments
Format Disk	Format only C and D. Formatting H or I will render your programmer inoperative until you restore the system software from floppy.
Duplicate Disk	Drives C, D, H, and I cannot be duplicated using the Duplicate Disk command.

**Backing up the MSM**

We STRONGLY suggest that you periodically make copies of the files on the C and D partitions of the MSM as backups. If the MSM were to "crash," you could restore your data files to C and D from your backups, and you could restore the system files from the Boot and Algorithm disks.

## 13. Install Devices

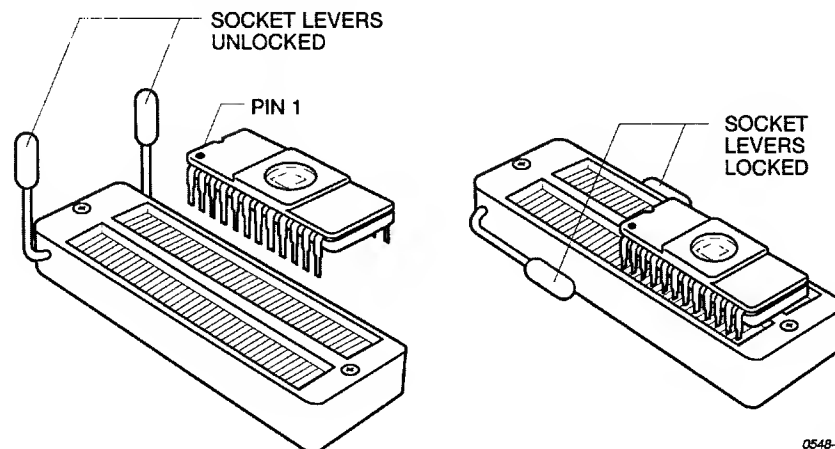
Follow the instructions below to install devices into the programmer's Module or Base sockets (see page 2-20 for a description of Bases). If the device type you wish to install is not listed in this section, refer to the documentation that is included with your Module or Base for installation instructions.

### Inserting a DIP Device

To insert a DIP device into the DIP socket, follow the steps below:

1. If a DIP Module or Base is not installed, install the Module or Base.
2. Unlock the socket by pulling up on the socket lever.
3. Insert the DIP device into the socket. Make sure that the device is bottom justified, as shown in Figure 2-10. If the device is not bottom justified, the programmer will not read or program the device.

**Figure 2-10**  
Inserting a DIP Device  
into the DIP Base



4. Lock the device into place by pressing the socket lever down.

### Installing a MatchBook

The MatchBook device carrier holds a device in place on the Base. When the device is locked into place, the conductive pad in the Base forms a conductive path between the pin drivers in the programmer and the device in the MatchBook.

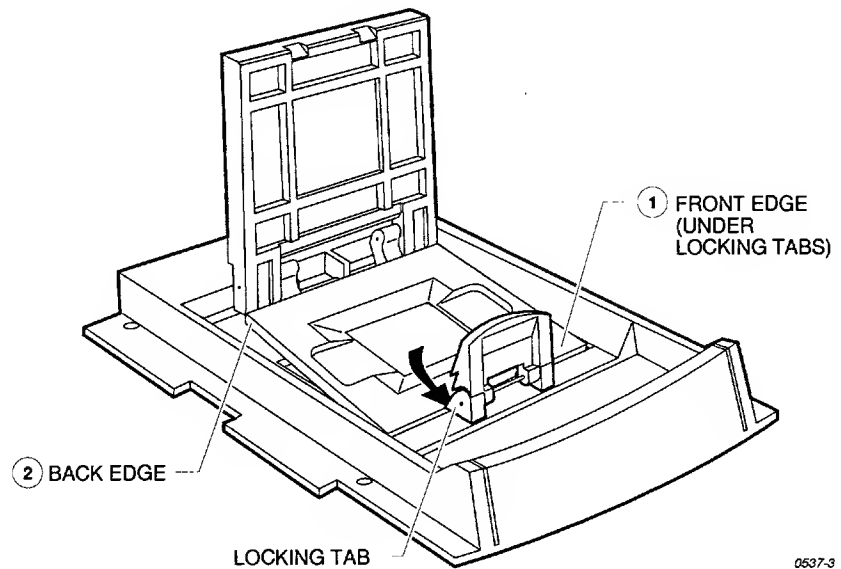
MatchBooks do away with clumsy and expensive sockets and adapters, and make inserting and removing surface mount devices easier and faster. MatchBooks eliminate the guesswork involved when you insert a device into a socket. All you have to do is align pin 1, set the device in the MatchBook, and close the lid.

The instructions below explain how to use a MatchBook and how to insert and remove a device from a MatchBook.

*Note: The information in this section covers all device types that are supported by MatchBook device carriers. For more information, see the section on the particular package type you are using.*

1. Insert the Base into the programmer. Lock the Base into place.
2. Select the MatchBook for the device you are going to use.
3. With the MatchBook open 90 degrees, insert it into the Base by setting the front edge onto the Base under the two locking tabs at the front edge of the Base. Then lower the back edge of the MatchBook into place on the Base. See Figure 2-11.
4. Insert the device into the MatchBook as described later in this section.

**Figure 2-11**  
*Inserting a MatchBook into the Base*

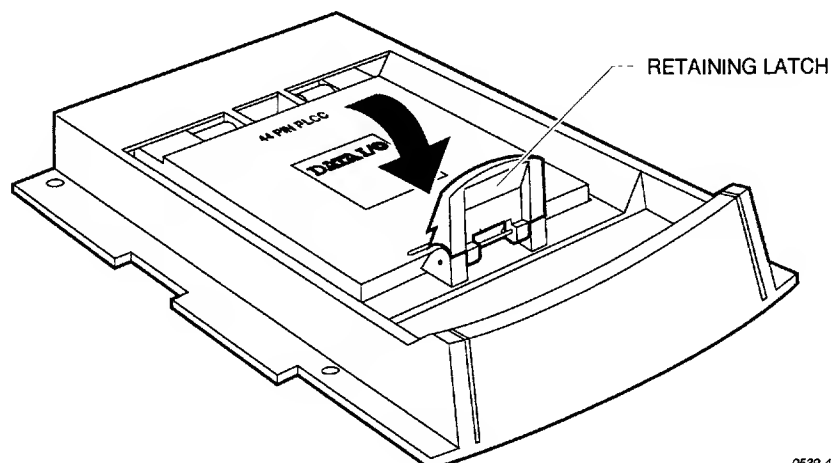


0537-3

5. Finally, close the MatchBook and press the retaining latch forward with your thumb until the latch snaps into place as shown in Figure 2-12.

**CAUTION:** Do not place excessive force on the top of the MatchBook, as this may cause premature wear on the conductive pad.

**Figure 2-12**  
*Closing the MatchBook*



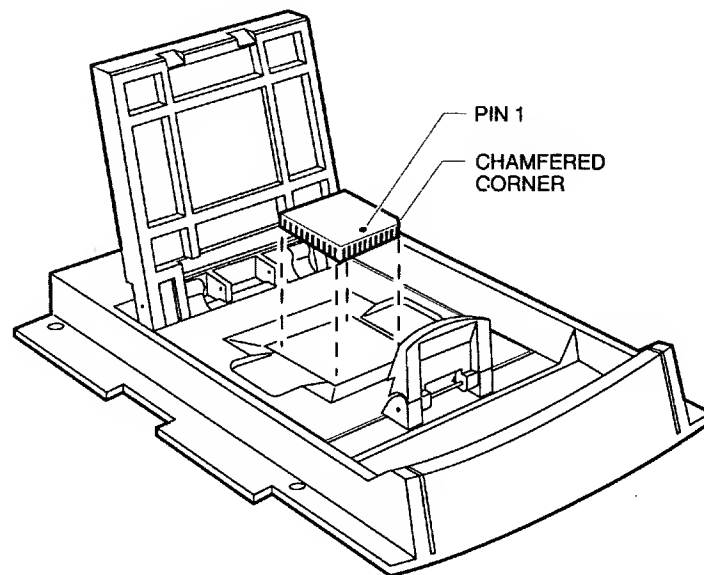
0539-4

## Inserting a PLCC or LCC Device into a MatchBook

Use the following procedure to insert a PLCC or LCC device into a MatchBook.

1. Select the appropriate MatchBook and Base. For example, if you are using a 44-pin PLCC device you would select the 44-pin PLCC MatchBook and the PLCC Base.
2. Insert the appropriate Base into the programmer.
3. Insert the appropriate MatchBook into the Base.
4. Position the PLCC or LCC device so that pin 1 is near the handle on the Base. There is a small dot molded into each MatchBook to help you align your device. Each PLCC MatchBook and LCC MatchBook also has a beveled corner to help you align devices with a chamfered corner to indicate pin 1. Figure 2-13 illustrates the proper positioning of a PLCC or LCC device.

*Figure 2-13*  
*Inserting a Device*  
*into the PLCC or LCC Base*



0638-3

5. Insert the device into the open MatchBook.
6. Finally, close the MatchBook and press the retaining latch forward with your thumb until the latch snaps into place, as shown in Figure 2-12.

---

**CAUTION:** *Do not place excessive force on the top of the MatchBook, as this may cause premature wear on the conductive pad.*

## Inserting an SOIC Device into a MatchBook

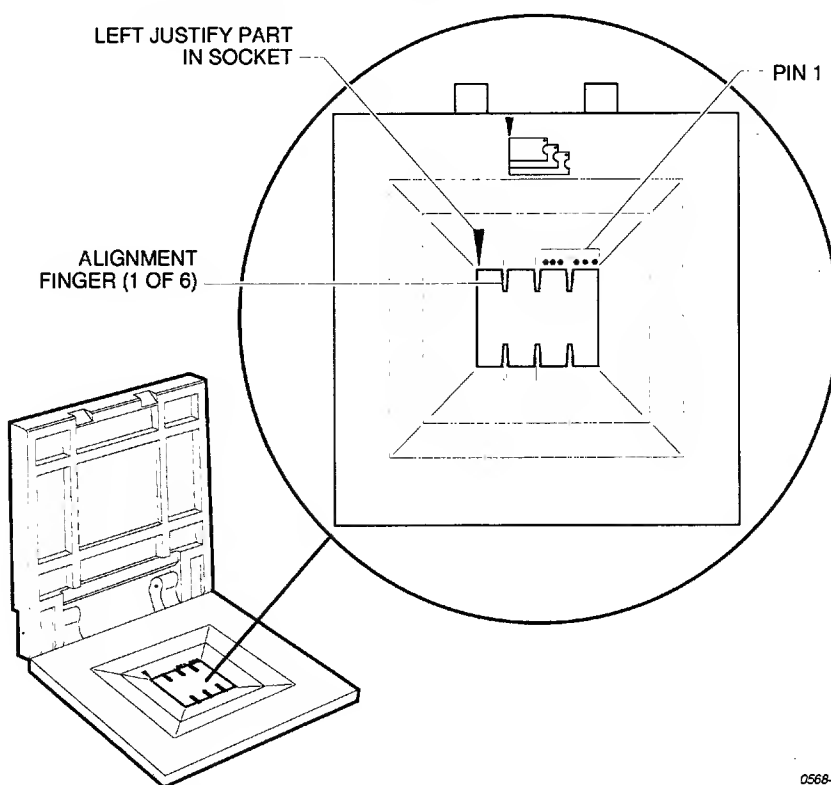
Use the following procedure to insert an SOIC device into the SOIC MatchBook. Before you go any further, make sure you have the SOIC Base properly installed in the programmer.

1. Select the appropriate MatchBook and Base. For SOIC devices, select the MatchBook according to package width: 0.150 inches, 0.300 inches, etc.
2. Insert the SOIC Base into the programmer. Insert the appropriate MatchBook into the SOIC Base.
3. Position the SOIC device so that pin 1 is up and to the right as you view it from the top.
4. Insert the SOIC device into the open MatchBook. Make sure the SOIC device is flush against the left-hand side of the MatchBook. Also make sure that the device is positioned between the six alignment fingers and not on top of them.

*Note: The device should be left justified. The unused portion of the socket will be on the right as you view it from the top.*

The small, round dots along the top of the opening, as shown in Figure 2-14, indicate the location of pin 1 for the various sizes of SOIC devices the SOIC MatchBook will accept.

**Figure 2-14**  
Inserting an SOIC Device



0568-3

5. Finally, close the MatchBook and press the retaining latch forward with your thumb until the latch snaps into place, as shown in Figure 2-12.



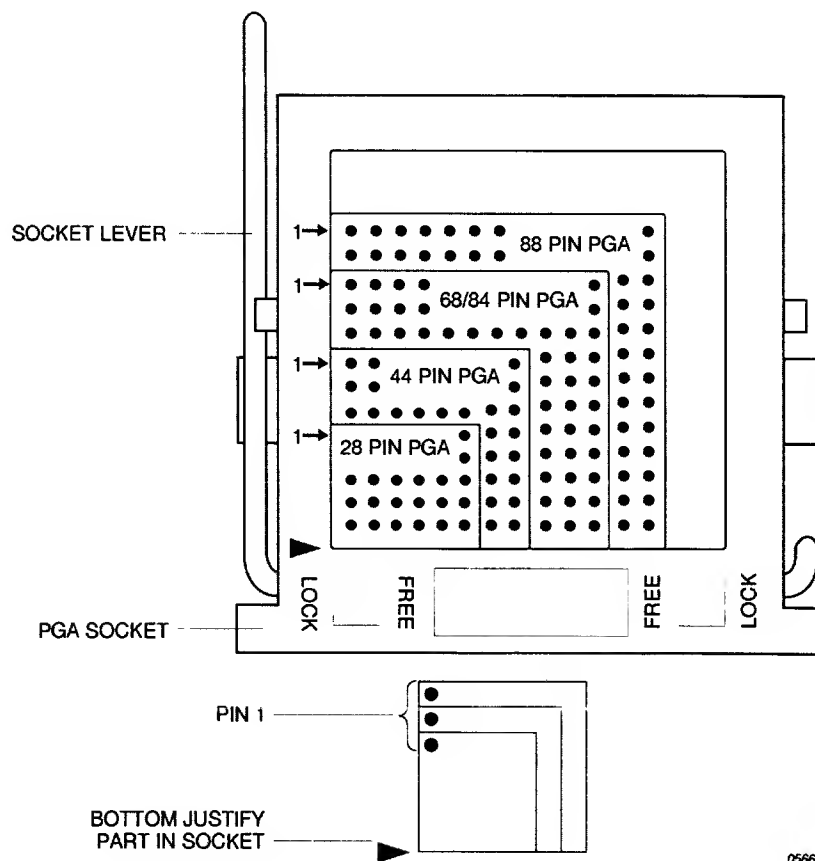
## Installing a PGA Device in a PGA Base

The PGA Base can accommodate PGA packages up to 15 x 15 pin arrays. Use the following procedure to insert a PGA device into the PGA Base.

*Note: When you insert a PGA device into the PGA Base, pay particular attention to the orientation and positioning of the device.*

1. Install the PGA Base into the programmer.
2. Unlock the PGA socket by lifting up on the socket lever.
3. Insert the device into the PGA socket. Make sure the PGA device is bottom justified and that pin 1 of the device is against the left side of the socket. Figure 2-15 shows the proper alignment of a PGA device.
4. Push the socket lever down to lock the PGA device into the PGA socket.

**Figure 2-15**  
Orienting a PGA Device in the PGA Base



0566-4

## Removing a Device From a MatchBook

To remove a device from the MatchBook, unsnap the retaining latch, open the MatchBook, and lift out the device.

## Special Note About the Conductive Pad

After a number of insertions you may notice an indentation in the middle of the conductive pad. (The conductive pad is the material the MatchBook rests on.) The indentation is normal and does not degrade the contact resistance and does not degrade the performance of the MatchBook.

The life of the pad is dependent on proper care as well as the pin count and package type of the device being used. Not all devices have the same tolerances and each device type may result in different life cycles for the pad.

If you experience an increase in device insertion errors or continuity errors, or if you experience a sudden drop in programming yields, it may be an indication that the pad needs to be replaced. The Base has been designed to allow you to replace the pads quickly and easily and to minimize downtime. Replacement pads are available from Data I/O.

## Pad Care

Each pad should be inspected and cleaned as needed; we recommend you do this approximately every 1000 insertions or once a month. It is normal for the pad to show signs of discoloration as it is used.

Clean the pad by blowing air over the pad. If you use compressed air, direct the air stream from the front or back of the Base.

---

**CAUTION:** *Blowing air from the side of the pad could lift the pad off the circuit board.*

To further clean the pad, apply a small amount of isopropyl alcohol on a cotton swab and, with a rolling motion, gently wipe off the pad. Make sure the pad is clear of any cotton filaments left over from the cleaning process.

---

**CAUTION:** *Do not use any petroleum or freon based products to clean the pad. These substances will cause premature deterioration of the pad material.*

## Maintenance

### Changing the Line Voltage

Data I/O has configured your UniSite to operate on 115 Vac unless specified otherwise. The line voltage indicator is visible through the window in the back panel door that covers the voltage selector wheel, which is shown in Figure 2-16. The ac line voltage that will be used to operate UniSite must match the number indicated in the window. If the voltage you need to use is NOT the same as the number in the window, follow the steps below to change the selected voltage.

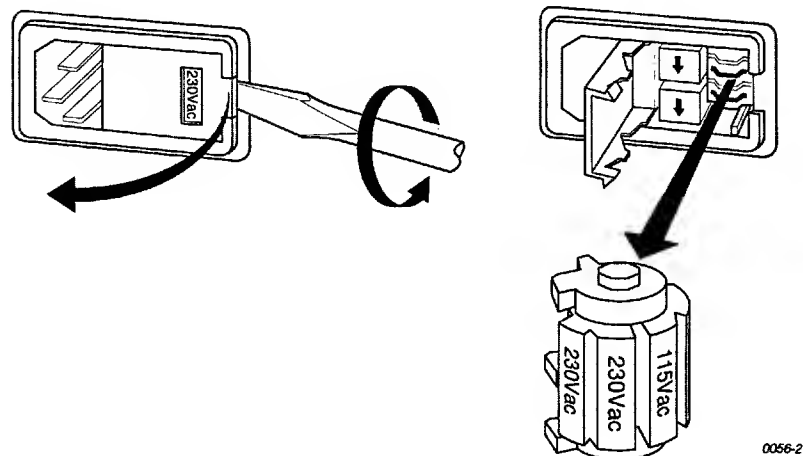
---

**CAUTION:** *You could damage UniSite if you operate it with the wrong line voltage.*

---

1. Disconnect the power cord.
2. Using a flat-tipped screwdriver, gently pry open the door that covers the voltage selector wheel.

**Figure 2-16**  
Removing the Voltage Selector Wheel



3. As shown in Figure 2-16, pull the voltage selector wheel out of its slot.
4. Rotate the selector until the correct line voltage points away from UniSite's rear panel. Insert the selector back into its slot.

---

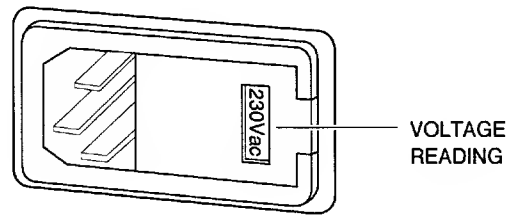
**Note:** *The voltage wheel has two positions: 115 Vac and 230 Vac. These are nominal voltages -- each voltage has high and low limits. The limits for 115 Vac is 90 Vac to 132 Vac and the limits for the 230 Vac position are 180 Vac to 264 Vac.*

---

5. Snap the door closed. The correct voltage now appears in the window.

6. Verify that the line voltage is correct by checking the line voltage indicator. Figure 2-17 shows the location of the voltage indicator.

**Figure 2-17**  
Voltage Reading



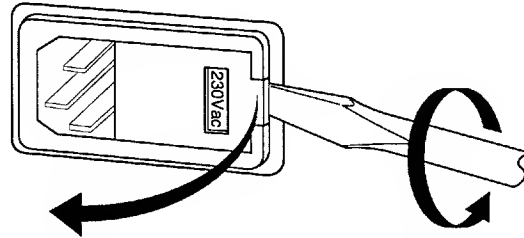
0057-2

## Replacing the Line Fuse

The line fuse is located behind the same door that covers the voltage selector wheel. Perform the following procedure to replace the line fuse. If the fuse is blown, replace it with one of the same size and rating.

1. Gently pry open the door that covers the fuse holder using a flat-tipped screwdriver.

**Figure 2-18**  
Opening the Fuse Holder

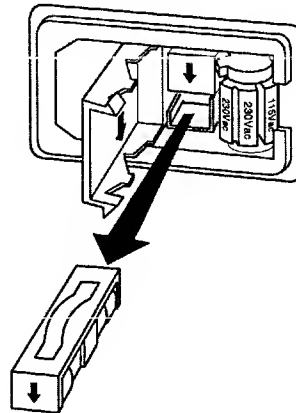


0058-2

*Note: The power entry module will accept two fuse cartridges. One cartridge holds USA standard size fuses (1/4" x 1 1/4") and the other holds international standard size fuses (5 mm x 20 mm). Only the bottom receptacle is connected to UniSite's circuitry.*

2. Pull the bottom fuse holder out of its slot.

**Figure 2-19**  
*Removing the Fuse Holder*



0059-2

3. Determine whether the fuse is intact. If it is intact, proceed to Step 4. If the fuse is blown, install a new fuse.

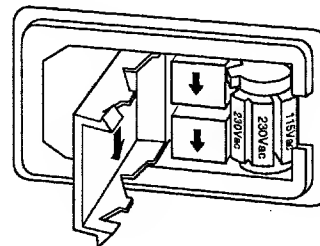
---

**CAUTION:** *For continued protection against the possibility of fire, replace only with a fuse of the correct voltage, current and type ratings.*

---

4. Insert the fuse holder into its slot so that the arrow points in the same direction as the arrows on the door of the fuse holder.
5. Snap the door closed.

**Figure 2-20**  
*Inserting the Fuse Holder*



0060-2

---

## What to Do Next Time

Next time you power up the programmer, you probably do not need to follow all the steps outlined in this chapter. Listed below are the normal steps for preparing for another session on the programmer.

---

*Note: If you have not used the programmer for awhile, or if you suspect the programmer might have been moved from one area to another, follow the procedure below before you use the programmer.*

To prepare the programmer for another session, follow the procedure below:

1. Check the power cords and cables between the programmer and the connected equipment.
2. If you are controlling the programmer from a PC or workstation, make sure it is on and that the terminal emulation software (such as HiTerm) is running.

If you are controlling the programmer from a terminal, make sure it is on.

If you are using PROMlink-6, select **VT100** on **Programmer Port** from the **Utilities** menu.

3. Insert the System disk into UniSite's disk drive A: (and insert an Algorithm disk into disk drive B: on two-disk drive systems).
4. Select and insert a Module (and Base if necessary) into the programmer. Make sure the Module and Base are locked in place.
5. Power up the programmer.
6. Verify the terminal type when the Power-on screen appears.

You are now ready to begin a new session on the programmer.

Refer to step 13, "Install Devices," on page 2-34 for instructions on how to install devices. Refer to Chapter 3, "Quick Start," for instructions on how to use the programmer's menu system (in terminal mode). Refer to Chapter 4, "Tutorial," for tutorials on how to use your programmer.



# 3 Quick Start

---

This chapter will help you begin operating UniSite. The list below shows you some of the major topics covered in this chapter.

- The UniSite Screen
- Controlling the Cursor
- Selecting and Executing a Command
- Getting Online Help
- Using Key Functions

---

## Before You Begin

Before you go any further in this chapter, make sure you have completed the setup and installation instructions in the previous chapter. You should also have read Chapter 1, "Introduction."

UniSite should be powered up and you should be looking at the Main Menu, shown in Figure 3-1.

*Figure 3-1*  
*The Main Menu*

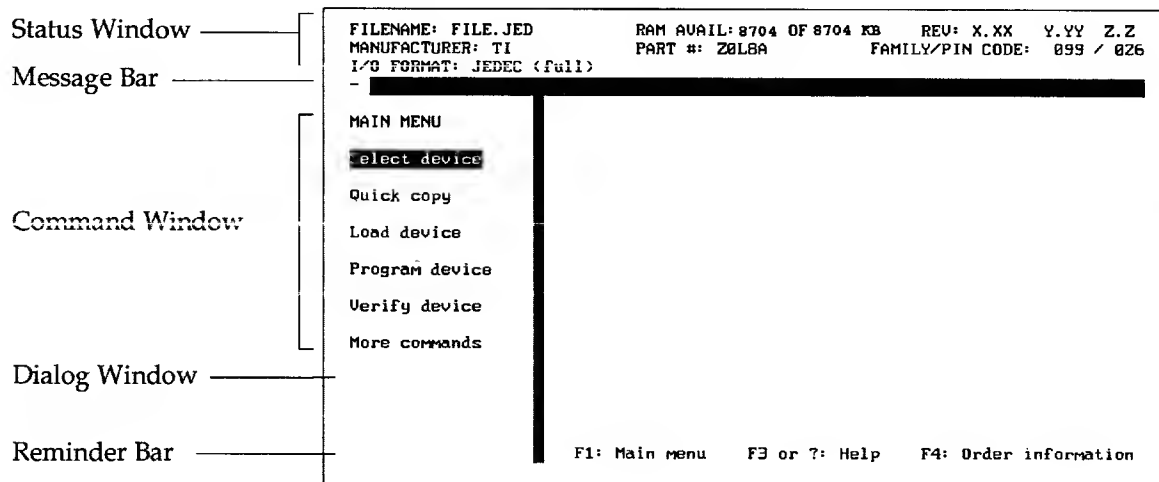
FILENAME:	RAM AVAIL: 128 OF 128KB	REV: X.XX Y.YY Z.Z
MANUFACTURER:	PART #:	FAMILY/PIN CODE: 000 / 000
I/O FORMAT:		
-		
MAIN MENU		
Select device		
Quick copy		
Load device		
Program device		
Verify device		
More commands		
F1: Main menu F3 or ?: Help F4: Order information		



## The UniSite Screen

You will see a consistent format to the screens as you use UniSite. Most screens, such as the screen shown in Figure 3-2, are broken into five areas: the status window, the message bar, the command window, the dialog window, and the reminder bar.

**Figure 3-2**  
*Areas of the UniSite Screen*



### Status Window

Occupying the top three lines of the screen, the status window displays important system information. The following are included in the status window:

- Name of the data file (FILE.JED in Figure 3-2)
- Amount of user RAM (8704 of 8704KB)
- Version numbers of the algorithm disk (X.XX), the system disk (Y.YY), and the system EPROM (Z.Z).
- Device manufacturer (TI) and part number (20L8A)
- Family code (099) and pinout code (026)
- Data translation format (JEDEC full)

### Message Bar

Located just below the status window, the message bar displays system and error messages. The action symbol is also located in the message bar. The action symbol rotates while an operation is taking place, indicating that UniSite is busy.

### Command Window

Occupying the left side of the screen is the command window. At the top of the window is the menu name, displayed in uppercase letters. Below the menu name, the available commands are displayed in upper- and lowercase letters.

---

<b>Dialog Window</b>	Occupying the largest window on the screen, the dialog window displays different information and system parameters, depending on the selected command.
<b>Reminder Bar</b>	Located at the bottom of the dialog window, the reminder bar tells you what function keys are available and what they will do if pressed.

---

## Moving the Cursor

Pressing the arrow keys moves the cursor in the direction indicated on the key. The cursor will wrap around when it has reached the edge (top, bottom, left, right) of a window.

---

## Selecting a Command

You can select a command using one of two methods: you can either press the first letter of the command, or move the cursor to the menu item and press **↓**.

---

## Accessing Online Help

Online Help screens are available throughout UniSite and provide both general help and context-sensitive help. Context-sensitive help gives you help text that is specific to a particular field on the screen. For example, if you are on a parameter selection screen, each parameter on the screen has a different piece of help text associated with it.

### Getting Help

To access help, move the cursor to the item you want help on and press either **F3** or **?**. The Help screen is divided into four sections: the key listing, the general help, the specific help, and the reminder bar.

See Figure 3-3 for a sample help screen.

### Key Listing

Displayed at the top of the help screen, the key listing provides a quick summary of some of the most often used key commands. Key combinations displayed with a dash between them, such as

CTRL-P

indicate you should press and hold the first key; then press the second key. Key combinations displayed with a space between them, such as

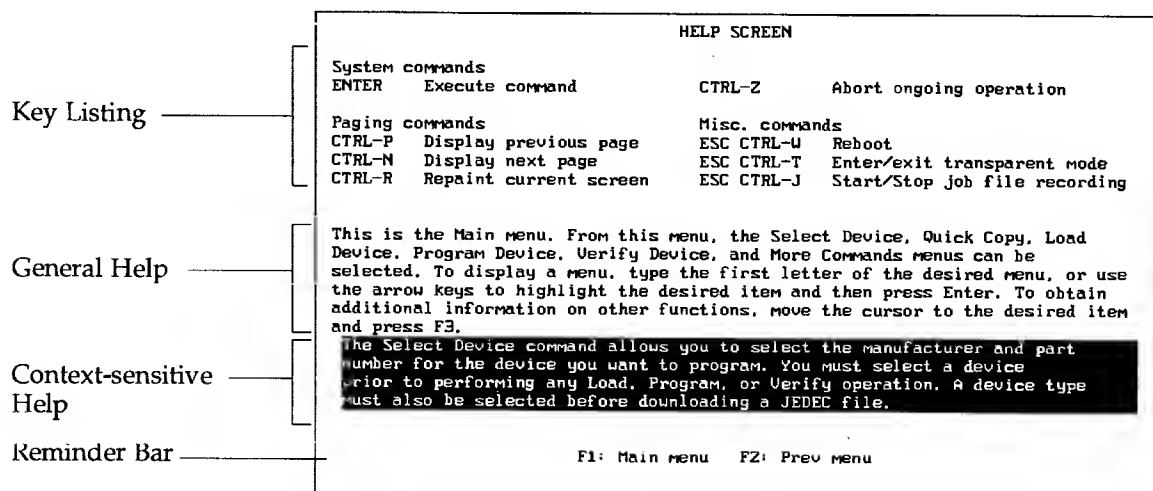
ESC CTRL-T

indicate you should press and release the first key, then press the key combination.

### General Help

Displayed in the middle of the screen, the general help text tells you about the next higher command or menu. Although the general help is not as specific as the context-sensitive help, it provides you with a good reference point.

**Figure 3-3**  
Areas of the Help Screen



#### Context-sensitive Help

Displayed in reverse video, the context-sensitive help provides you with information about the specific item the cursor was on when you called the help function. The information in this field changes every time you move the cursor to a different location on the screen.

#### Reminder Bar

Displayed on the bottom line of the screen, the reminder bar shows which function keys you can use to exit Help.

### Accessing Online Help for System Messages

Online help is available for non-fatal system messages, which result from situations that do not interrupt UniSite's operation. Fatal messages, which result from situations that do interrupt UniSite's operation, are listed and described in Chapter 8, "Messages."

Non-fatal error messages are generally displayed in the message bar. To access the online help for the message, press **F3** or **?**. UniSite displays the online help for the message. Exit the message help screen as you would any help screen.

### Accessing Device-Specific Online Information

After selecting a device, you will see the following message in the message bar if there is online device-specific information for the selected device.

Hit F3 or ? to view device specific message.

Press **F3** or **?** to display the device-specific information. If there is more than one screen of device-specific information, press **Ctrl** + **N** to view the next screen of information. Exit the screen as you would any help screen.

## Exiting Help

To exit any Help screen, press either **F1** or **F2**. **F1** returns you to the Main Menu and **F2** displays the previous screen from which the Help function was invoked.

## Using Key Functions

Some of UniSite's functions may be performed by pressing a key or a combination of keys. When using the **Ctrl** key, hold it down and then momentarily press the second key. The key functions are listed below with their corresponding keystroke sequence.

Keystrokes	Description
<b>F1</b>	Return to the Main Menu
<b>F2</b>	Go to the previous menu
<b>F3</b> or <b>?</b>	Display online help for the current menu and cursor position
<b>F4</b>	Display the Optional Parameters screen
<b>↓</b>	Execute highlighted command
<b>Space</b>	Toggle a parameter
<b>Ctrl</b> + <b>N</b>	Display next page
<b>Ctrl</b> + <b>P</b>	Display previous page
<b>Ctrl</b> + <b>R</b>	Repaint screen
<b>Ctrl</b> + <b>Z</b>	Halt current operation
<b>Esc</b> <b>Ctrl</b> + <b>T</b>	Enter/exit transparent mode with host computer
<b>Esc</b> <b>Ctrl</b> + <b>J</b>	Start/stop job file recording
<b>Esc</b> <b>Ctrl</b> + <b>W</b>	Restart UniSite (warm boot)
<b>Break</b> <b>A</b>	Execute AutoBaud



# 4 *Tutorial*

---

---

## Before You Begin

Before you start the Sessions in this chapter, read Chapter 1, "Introduction," which introduces you to UniSite. Next, read Chapter 2, "Setup and Installation," which tells you how to set up UniSite and install the software.

Finally, read Chapter 3, "Quick Start," which gives you a quick tour of UniSite and introduces you to UniSite's interface. Make sure UniSite is connected properly and is working before you start the Sessions.

---

## Should I Read This Chapter?

You should read and follow the Sessions in this chapter if you are unfamiliar with UniSite Universal Programmer or if you are unfamiliar with device programmers in general. By following the Sessions, you will learn how to select a device, load device data, edit data, program the device, and verify that it programmed correctly.

The Sessions will not teach you everything about UniSite; instead, they will give you a working knowledge of UniSite. Chapter 5, "Commands," contains detailed information about the features and capabilities of UniSite. Refer to Chapter 5 when you want more information about a particular command or procedure.

---

## What Will I Learn?

The Sessions are meant to be read sequentially. The most basic steps are taught first and lead to more complex procedures. The Sessions are organized as follows:

Session 1: Navigating Through the Menus

Session 2: Selecting a Device

Session 3: Selecting a Keep Current Algorithm

Session 4: Loading Data from a Device

Session 5: Loading Data from Disk

Session 6: Selecting a Translation Format

Session 7: Loading Data from a PC Using HiTerm

Session 8: Loading Data from a Host

Session 9: Editing Data

Session 10: Programming a Memory Device

Session 11: Verifying a Device

Summaries at the end of each Session remind you of the commands and procedures you are learning.

---

## Review of Programming Concepts

### Select Device

Before attempting most operations, you must first tell UniSite what device you will be working with. UniSite then selects the appropriate programming algorithm based on the manufacturer and device part number selected. Once you have selected a device, you can move on to other device operations, such as loading, programming, and verifying.

### Load Data

Load operations determine how you move device data into UniSite. You can load data from a device, from one of UniSite's internal disk drives, or from one of the two serial ports on UniSite (for example, from the Remote port).

### Program Device

Programming transfers the device data into the device inserted in UniSite. The programming is done according to the programming algorithm selected by the Select Device command. The programming operation also includes a Verify Operation, which is described below.

### Verify Device

A verify operation compares the data in a programmed device against the data in UniSite's RAM or against the data in a disk file. In the case of logic devices, verifying can also include functional testing. Verify is normally an automatic part of the program operation, but additional verify operations can provide useful information about many programming errors.

## Session 1: Navigating Through the Menus

This Session describes the organization of UniSite's menus and gives you an opportunity to explore the menu tree and the online help.

### Select a Menu Item

First, let's go over how you select an entry from the command window. (Remember, the command window is on the left side of the vertical bar.)

To select a command, either move the cursor to the menu item and press **[↓]** or press the first letter of the menu item.

For example, if you want to test your UniSite, select the Self-test command on the More Commands menu. To access the More Commands menu, move the cursor to the More Commands menu item and press **[↓]**. Or, as mentioned above, simply press **[M]**.

The More Commands menu should appear. It should look like the screen shown in Figure 4-1.

**Figure 4-1**  
*The More Commands Menu Screen*

FILENAME:	RAM AVAIL: 128 OF 128KB	REV: X.XX	Y.YY	Z.Z
MANUFACTURER:	PART #:	FAMILY/PIN CODE: 000 / 000		
I/O FORMAT:				
MORE COMMANDS				
Configure system				
Device checks				
Edit data				
File operations				
Job file				
Remote control				
Self-test				
Transfer data				
Yield tally				
F1: Main menu		F2: Prev menu		F3 or ?: Help

If a different menu appears, don't worry. Press **[F1]** and try again.

Scan the menu, which is arranged alphabetically, and find the Self-test menu item. To access the Self-test, press **[↓]** to move the cursor to the Self-test menu item and then press **[↓]** to execute the command. (Remember, you could also have pressed the first letter of the menu item to select that command. In this case, you would press **[S]**.)

The dialog window should fill with the Self-test screen and the top of the window should read

SYSTEM DIAGNOSTIC TESTS



You have reached the Self-test screen, which is shown in Figure 4-2.

**Figure 4-2**  
*The Self-test Screen*

FILENAME:		RAM AVAIL: 128 OF 128KB		REV: X.XX Y.YY Z.Z	
MANUFACTURER:		PART #:		FAMILY/PIN CODE: 000 / 000	
I/O FORMAT:					
-					
MORE COMMANDS		SYSTEM DIAGNOSTIC TESTS			
Configure system	Waveform board	PASS	Pin Control Unit	PASS	
Device checks	EPROM	PASS	Serial ports	PASS	
Edit data	System RAM	PASS	User RAM	PASS	
File operations	Disk A	PASS	Disk B	PASS	
Job file			PSM	PASS	
Remote control			FSM	---	
<b>Self-test</b>					
Transfer data					
Yield tally					
		Pin Driver board number 5 10 15			
		PPPPP PP --- --			
		[P]Pass [F]Fail [?]Untested [-]Not installed			
		Perform All Tests Test mode ONE PASS			
		Return: Execute			
		F1: Main menu F2: Prev menu F3 or ? : Help			

## Select Online Help

To access the context-sensitive, online help for the item the cursor is highlighting, press either **[F3]** or **[?]**. The entire screen clears and fills with the Help screen. The top third of the Help screen is a list of commonly used keys, the next third is general help, and the last third is the context-sensitive help.

**Figure 4-3**  
*Areas of the Help Screen*

		HELP SCREEN	
Key Listing	System commands		
	ENTER Execute command	CTRL-Z	Abort ongoing operation
	Paging commands	Misc. commands	
	CTRL-P Display previous page	ESC CTRL-W	Reboot
General Help	CTRL-N Display next page	ESC CTRL-T	Enter/exit transparent mode
	CTRL-R Repaint current screen	ESC CTRL-J	Start/Stop job file recording
Context-sensitive Help	This is the Main menu. From this menu, the Select Device, Quick Copy, Load Device, Program Device, Verify Device, and More Commands menus can be selected. To display a menu, type the first letter of the desired menu, or use the arrow keys to highlight the desired item and then press Enter. To obtain additional information on other functions, move the cursor to the desired item and press F3.		
	The Select Device command allows you to select the manufacturer and part number for the device you want to program. You must select a device prior to performing any Load, Program, or Verify operation. A device type must also be selected before downloading a JEDEC file.		
Reminder Bar	F1: Main menu F2: Prev menu		

## Context-sensitive Help

Context-sensitive help means that the text displayed on the screen changes every time you move the cursor to a different field. If the cursor is in the Disk field you get information on the Disk; if the cursor is in the RAM field, you get information on the RAM.

To leave the help screen and return to the Self-test screen, press **[F2]**. (Remember, if you forget what key to press, look at the reminder bar on the bottom of the screen for a quick reminder.) You should now be looking at the Self-test screen.

To illustrate that the context-sensitive help changes every time you change the cursor position, get help for two different fields on the Self-test screen. For example, move the cursor to the RAM field and access online help. (Remember, press **F3** or **?** to get help and **F2** to exit the help screen.)

Next, move the cursor to the Disk field and access the online help. (Remember, press **F3** or **?**.)

Notice that the context-sensitive help (the text in reverse video on the lower third of the screen) changed. The general help (the text in normal video on the middle third of the screen) didn't change because you are still on the Self-test screen. The general help changes when you move to another menu level. For example, leaving the Self-test screen and returning to the More Commands menu causes the general help to change.

To change the general help, press **F2** until the flashing cursor returns to the command window. Then access the online help. Notice that the general help has changed.

## Exit the Online Help

In the previous section, you pressed **F2** to return to the Self-test screen from the help screen. Pressing **F2** takes you one step closer to the Main Menu.

If you had wanted to return to the Main Menu quickly, you could have pressed **F1** instead.

Let's try it. If you are not already at the Main Menu, press **F1** to return to the Main Menu. Press **M** and then **S** to access the Self-test screen. From here, you can either press **F2** twice to return to the Main Menu, or you can press **F1** once to return to the Main Menu. Step through the menus until you are comfortable. (Access the online help if you want to.)

When finished, return to the Main Menu.

This completes your tour of UniSite's user interface and online help system.

## Review

In this Session you learned how to navigate through the UniSite interface.

Select menu items by either pressing the first letter of the command or moving the cursor to the command and pressing **J**.

To access previous screens, press **F2**. To return to the Main Menu, press **F1**.

Finally, you learned how to access online help. To access the online help screens, either press **F3** or **?**.

## Session 2: Selecting a Device

This Session describes how to tell UniSite the manufacturer and part number of the device you are using. The device selection process is a two-step process: first you select the device manufacturer, and then you select the device part number.

Read this Session if you are going to select a device that is supported by an algorithm included on one of the UniSite Algorithm disks. The next Session, "Selecting a Keep Current Algorithm", covers how to select a device that is supported by a Keep Current Express algorithm.

For more information on the Keep Current Express Subscription Service, see the documentation behind the Keep Current tab at the back of this binder.

### Before You Begin

You should have completed Session 1, which introduces you to the UniSite interface. Also, make sure that you are at the Main Menu before you start. (Press **F1** to return to the Main Menu.)

### Can I Use Another Device?

If you do not have an AMD 27256 (the device we are using for this Tutorial), substitute the manufacturer and part number of the device you want to use. Keep in mind that the device you are using might not have the same capabilities as the AMD 27256. For example, the AMD 27256 supports Electronic ID while the Hitachi 27256 does not. Also, be sure to insert the correct algorithm disk.

### Select a Manufacturer

Insert Algorithm Disk 2 (for memory or Emicro devices) into drive B.

Choose the Select Device command from the Main Menu. (You can either press **S** or move the cursor to the Select Device menu item and press **J**.) The Manufacturer List screen appears, and should look like the screen shown in Figure 4-4.

**Figure 4-4**  
*The Device Manufacturer Selection Screen*

```

FILENAME:                               RAM AVAIL: 128 OF 128KB   REV: X.XX  Y.YY  Z.Z
MANUFACTURER:                           PART #:               FAMILY/PIN CODE: 000 / 000
I/O FORMAT:
/

```

MANUFACTURER LIST				Page 1 of 2
(1) FAM/PINCODE	(14) Goldstar	(27) Mit-Plastics	(40) Ricoh	
(2) KEEP CURRENT	(15) Gould	(28) MMI-PROM	(41) Rockwell	
(3) Actel	(16) Harris	(29) MMI-LOGIC	(42) Samsung	
(4) Altera	(17) Hitachi	(30) Motorola	(43) Seeg	
(5) AMD	(18) Hyundai	(31) National	(44) SGS-Thomson	
(6) AMD-XPGM	(19) ICT	(32) Natnl-XPGM	(45) SGS-Tho-XPGM	
(7) Asahi Kasei	(20) IDT	(33) NEC	(46) Sharp	
(8) Atmel	(21) Intel	(34) Oki	(47) Signetics	
(9) Catalyst	(22) Intel-XPGM	(35) Omni-Wave	(48) Seiko Instr	
(10) Cypress	(23) Lattice	(36) Panasonic	(49) Simtek	
(11) Exel	(24) Microchip	(37) PLUS Logic	(50) SMS	
(12) Fairchild	(25) Mikroelek	(38) PLX	(51) Sony	
(13) Fujitsu	(26) Mitsubishi	(39) Raytheon	(52) TI	

```

Manufacturer: 1 Device Type: All Mode: Single device
^N: Next page ^P: Prev page
F1: Main menu F2: Prev menu F3 or ? : Help

```

Examine the list of manufacturers and notice that they are listed alphabetically. Also, notice that some manufacturers are listed by their commonly used abbreviations. For example, Advanced Micro Devices is listed as AMD and Texas Instruments is listed as TI.

### Expanding Your Options

Look at the manufacturer screen again. Specifically, look at the upper-right corner of the screen and notice the text that looks similar to the following:

Page 1 of 2

This tells you how many screens (pages) of manufacturers there are and what page you are on. To go to the previous page of manufacturers, press **Ctrl** + **P**. To go to the next page of manufacturers, press **Ctrl** + **N**. When the screen repaints, you see a new list of manufacturers and the page counter now reads

Page 2 of 2

Keep paging through the screens until you find the manufacturer you are looking for. For this Session, we will be using a device from AMD. Page through the manufacturer listing until you see the entry for AMD.

### The Device Type Filter

The Device Type filter allows you to select which device types you want displayed. Press **Space** to cycle through the three settings: All, Memory & Emicros, or Logic Only. When you select a manufacturer, UniSite displays only those devices that fit the filter you selected.

For this Session, we will be using a 27256, which is an EPROM. Move the cursor to the Device Type field, press **Space** to cycle through the device types until Memory & Emicros appears in the field. Notice how the display changes when you cycle from one filter to another.

### The Programming Mode Filter

The Mode filter toggles UniSite between Single Device programming and Gang/Set programming. Gang programming is the programming of a single data file into multiple devices at once. Set programming is the partitioning and programming of a single large data file into multiple devices.

Move the cursor to the Mode field and press **Space**. Watch what happens to the Manufacturer List when you toggle from Single to Gang/Set: UniSite displays fewer manufacturers. This is because some of the manufacturers displayed in Single Mode do not make devices that can be programmed in Gang or Set Mode. As with the Device Type field, UniSite only displays the manufacturers that make devices that fit the currently selected filters.

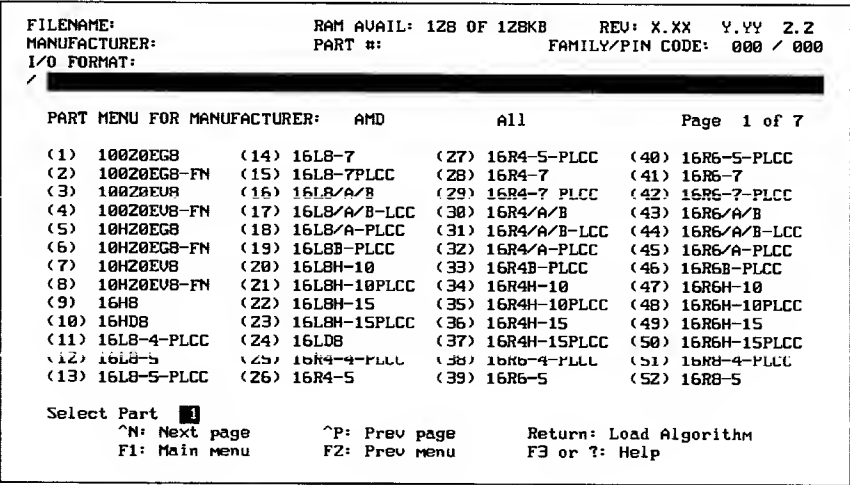
For this Session, we will be programming a single 27256 EPROM. So, set the Mode filter to Single. However, if we wanted to make several copies of the EPROM, you would set the Mode filter to Gang/Set. But, for this Session, set the Mode filter to Single.

Selecting the Manufacturer

For this Session, we will be using an AMD 27256 DIP EPROM. To select AMD from the Manufacturer List, move the cursor to the Manufacturer field, enter the number shown to the left of the name, and press **[J]**.

After selecting the manufacturer, the Part Menu should appear, and should look like the screen shown in Figure 4-5.

Figure 4-5  
The Part Number Selection  
Screen



Select a Device Part  
Number

Selecting a device part number is the same as selecting a device manufacturer: find the item in the menu and enter the number beside the item. In this case we are looking for a 27256.

Where Is It?

Notice that the screen is full of part numbers, and that the part numbers are arranged alphanumerically. If you examine the screen, you might find that the 27256 is not listed. This is because all the supported devices for this manufacturer could not be shown on one screen.

Expanding Your Options

Look at the screen again. Specifically, look at the upper-right corner of the screen and notice the text that looks similar to the following:

Page 1 of 5

This tells you how many screens (pages) of devices there are and what page you are on. To go to the next page, press **[Ctrl] + [N]**. When the screen repaints, you see a new list of devices and the page counter now reads

Page 2 of 5

Keep paging through the screens until you find the 27256 part number.

## Reviewing Your Options

If you scroll too far forward, and want to look at a previous screen, press **Ctrl** + **P** until you are back to where you want to be.

### Found It

After finding the 27256 on the part menu, type the number enclosed in parenthesis. For example, in Figure 4-5, you would type **7** **↓** .

UniSite loads the algorithm for the device you selected, in this case an AMD 27256. While UniSite is loading the algorithm, the action symbol rotates. (Remember, the action symbol is located at the left end of the message bar at the top of the screen.)

### Problems?

If you try to select a device before properly updating a new version of software, UniSite displays the following message:

OPERATION ABORTED: Product security violation

If you are using a new version of software (i.e., you received an update) for the first time, you must perform the Update operation prior to selecting a device command. See the description of the Update command in Chapter 5 for more information.

---

*Note: You do not have to run the Update command on an "out of the box" UniSite. However, you do need to run the Update command on any subsequent updates to the software.*

If you didn't receive any error messages when you selected the AMD 27256, then don't worry: your system software is installed properly. Continue with this Session.

### Done

When UniSite has loaded the programming algorithm for the device you have selected, UniSite will return to the Main Menu. The manufacturer and part number of the device you selected will appear in the status window. At this point, you will see only screens related to the type of device you selected. For example, if you choose a logic device, you will see only screens that are required to load, program, edit, and verify a logic device.

## Accessing Device-specific Online Information

Some devices have special information relating to their use and programming. The special information is listed in the Device List as footnotes and is also available online. If you select a device that has footnotes or other device-specific information, UniSite displays the following message in the message bar:

Hit F3 or ? to view device specific message.

To view the online information, press **F3** or **?**. The screen clears and UniSite displays the device-specific information. If there is more than one screen of device-specific information, press **Ctrl** + **N** to view the next screen of information. Press **F2** to exit the help screen.

For this Session, the device you have selected does not require any online information. However, if you had selected a different device, a Lattice 22V10 for example, you would have seen the message described above.

## Review

Selecting a device is a two-step process

- First, select the manufacturer of the device.
- Second, select the part number of the device.

Some devices have special information related to their use and programming. The information is available online and can be viewed after a device is selected. When prompted, press **F3** or **?** to view the information.

## Session 3: Selecting a Keep Current Algorithm

This Session describes how to select a device that is supported by a Keep Current algorithm you downloaded from the Keep Current Bulletin Board System (BBS). The device selection process is a two-step process: first UniSite displays a list of the available Keep Current algorithms, and then you select the algorithm.

Read this Session if you are going to select a device that is supported by a Keep Current algorithm. The previous Session, "Selecting a Device," covers how to select a device that is supported by an algorithm that is included on the UniSite Algorithm disk.

### Before You Begin

You should have completed Session 1, which introduces the UniSite interface. Also, make sure that you are at the Main Menu before you start. (Press **F1** to return to the Main Menu.)

Also, you should have placed your Keep Current algorithm(s) on a 3.5-inch disk formatted in your UniSite. See the Keep Current documentation for more information.

### Can I Use Another Device?

If you have not downloaded a Keep Current algorithm for your UniSite, you will not be able to complete this Session. However, you can still read the remainder of the Session to learn more about how to select a Keep Current algorithm. Or, you can skip this Session and continue with Session 4.

If you select a device supported by a Keep Current algorithm, keep in mind that the device you are using might not have the same capabilities as the device we are using for the rest of this Tutorial.

### Select the Keep Current Option

First, choose the Select Device command from the Main Menu. (Either press **S** or move the cursor to the Select Device menu item and press **↓**.) The Manufacturer List screen appears, as shown in Figure 4-6.

**Figure 4-6**  
The Device Manufacturer Selection Screen

```

FILENAME:          RAM AVAIL: 128 OF 128KB   REV: X.XX  Y.YY  2.2
MANUFACTURER:      PART #:                   FAMILY/PIN CODE: 000 / 000
I/O FORMAT:
/

```

MANUFACTURER LIST				Page 1 of 2
(1) FAM/PINCODE	(14) Goldstar	(27) Mit-Plastics	(40) Ricoh	
(2) KEEP CURRENT	(15) Gould	(28) MMI-PROM	(41) Rockwell	
(3) Actel	(16) Harris	(29) MMI-LOGIC	(42) Samsung	
(4) Altera	(17) Hitachi	(30) Motorola	(43) Seeg	
(5) AMD	(18) Hyundai	(31) National	(44) SGS-Thomson	
(6) AMD-XPGM	(19) ICT	(32) Natnl-XPGM	(45) SGS-Tho-XPGM	
(7) Asahi Kasei	(20) IDT	(33) NEC	(46) Sharp	
(8) Atmel	(21) Intel	(34) Oki	(47) Signetics	
(9) Catalyst	(22) Intel-XPGM	(35) Omni-Wave	(48) Selko Instr	
(10) Cypress	(23) Lattice	(36) Panasonic	(49) Simtek	
(11) Exel	(24) Microchip	(37) PLUS Logic	(50) SMS	
(12) Fairchild	(25) Mikroelek	(38) PLX	(51) Sony	
(13) Fujitsu	(26) Mitsubishi	(39) Raytheon	(52) TI	

```

Manufacturer: 1 Device Type: All Mode: Single device
^N: Next page ^P: Prev page
F1: Main menu F2: Prev menu F3 or ? : Help

```



Examine the list of manufacturers and locate the KEEP CURRENT entry. All the Keep Current algorithms are located under the KEEP CURRENT menu.

### Expanding Your Options

Look at the manufacturer screen again. Specifically, look at the upper-right corner of the Manufacturer List screen and notice the text that looks similar to the following:

Page 1 of 2

This tells you how many screens (pages) of manufacturers there are and what page you are on. If you do not see the KEEP CURRENT entry, press **Ctrl** + **N** to display the next page of manufacturers. (Press **Ctrl** + **P** to display the first page of manufacturers.) When the screen repaints, you will see a new list of manufacturers and the page counter will read

Page 2 of 2

Page through the screens until you find the KEEP CURRENT entry.

### The Device Type Filter

The Device Type filter has no effect on the devices displayed when you select Keep Current devices.

### The Programming Mode Filter

The Mode filter has no effect on the devices displayed when you select Keep Current devices. However, setting the Mode filter to Gang/Set will enable gang/set programming for devices that support that programming mode. Press **Space** to toggle between Single Device operations and Gang/Set operations.

### Insert the Keep Current Algorithm Disk

Before you select the KEEP CURRENT entry, make sure the 3.5-inch disk containing the Keep Current algorithm(s) is inserted in drive B. If you are using a single-drive UniSite, insert the disk in drive A.

### Selecting the Manufacturer

For this Session, we will be selecting a Keep Current algorithm that you have downloaded from the Data I/O Keep Current BBS. Regardless of manufacturer, all Keep Current algorithms are located under the KEEP CURRENT entry.

To select KEEP CURRENT from the Manufacturer List, move the cursor to the Manufacturer field, enter the number shown to the left of the entry and press **Enter**. For example, in Figure 4-6 you would type **2** **Enter**.

What do you see?

After selecting the KEEP CURRENT entry, the Keep Current Part List screen will appear, as shown in Figure 4-7.

**Figure 4-7**  
The Keep Current Part List Screen

```

FILENAME:          RAM AVAIL: 128 OF 128KB  REV: X.XX  Y.YY  Z.Z
MANUFACTURER:      PART #:                  FAMILY/PIN CODE: 000 / 000
I/O FORMAT:
-

```

---

```

                                KEEP CURRENT PART LIST                Page 1
Call Keep Current BBS at 1-206-881-3465 for latest KEEP CURRENT part list

```

	Manufacturer	Part	Alg. Rev.	File name
(1)	AMD	CEZZV10-UES-PLCC	3.7a	A:AMD.KC1
(2)	AMD	1020EG8-FN	3.7b	A:AMD.KC2
(3)	AMD	16R4H-15	3.7a	A:16R4H_15.KC3
(4)	ALT	910T-PLCC	3.7b	B:910T.KC1
(5)	EXEL	46C15	3.7a	B:46C15.KC1
(6)				
(7)				
(8)				
(9)				
(10)				

```

Select Part 1

^N: Next page      ^P: First page
F1: Main menu      F2: Prev menu      F3 or ? : Help

```

If you see the Keep Current Part List screen, continue with the section titled "Select the Keep Current Algorithm."

If you do not see the Keep Current Part List screen, you are probably looking at a screen that describes the Keep Current Express service. UniSite will not display the Keep Current Part List screen if you do not have any Keep Current algorithms.

If you have downloaded some Keep Current algorithms, make sure the disk with the algorithms is in drive B. If you are using a single-drive UniSite, insert the disk in drive A. After you insert the disk with the Keep Current algorithms, press **[F1]** and restart this Session.

## Select the Keep Current Algorithm

At this point, you should be looking at the Keep Current Part List screen, which is shown in Figure 4-7.

Selecting a Keep Current algorithm is the same as selecting a standard device algorithm: find the item in the menu and enter the number corresponding to the item. In this case you are looking for a Keep Current algorithm you downloaded from the Keep Current BBS.

## Where Is the Algorithm?

Notice that the Keep Current Part List screen lists both the device manufacturer and the device part number. Also, notice that the devices are listed in no particular order: UniSite lists the devices in the order that they are found on the disk.

If you do not see the algorithm you are looking for, make sure the disk with the Keep Current algorithm is inserted in UniSite. If you have more than 10 algorithms on one disk, the algorithm you are looking for could be on the next screen of algorithms. Press **[Ctrl] + [N]** to display the next page of algorithms.

Keep paging through the screens until you find the algorithm you are looking for.

**Reviewing Your Options**

If you page too far, press **Ctrl** + **P** to return to the first page of Keep Current algorithms.

**Found It**

After finding the algorithm, type the number enclosed in parentheses and press **↓**.

UniSite will load the algorithm for the device you selected. The action symbol rotates while UniSite is loading the algorithm.

After loading the algorithm, UniSite returns to the Main Menu and updates the status window to show the device you have selected.

**Keep Current Algorithms and Software Updates**

Each Keep Current algorithm is designed to work with a particular version of system software. When UniSite displays the available Keep Current algorithm(s) on the Keep Current Part List screen, it filters out the Keep Current algorithms that are invalid and incompatible with the installed version of system software.

A Keep Current algorithm and a version of UniSite system software are compatible when the numbers to the left and immediate right of the decimal point match. For example,

Algorithm Version	System Software Version	Compatible?
3.51	3.5	Yes
3.7	3.7	Yes
3.6	3.7	No

Keep Current algorithms are valid for one major release of software because the Keep Current algorithms are included with the next release of system software.

The following example illustrates a typical Keep Current scenario:

- In May, you update your system software to version X.4. At the same time, you enroll in the Keep Current Express Subscription Service.
- In June, Cruft Technologies announces a new device, the Cruft 1263.
- A week later, Data I/O announces support for the Cruft 1263 and places a Keep Current algorithm for the Cruft 1263 on the Keep Current BBS.
- The next day, you call the Keep Current BBS and download the new algorithm for the Cruft 1263.
- In August, Data I/O releases version X.5 system software, complete with the new algorithm for the Cruft 1263.
- You update your UniSite to version X.5 system software. The algorithm for the Cruft 1263 is part of the system software.

With Keep Current algorithms, you get immediate device support, rather than having to wait for the next release of system software.

---

## Special Note

The next four Sessions cover how to load programming data into UniSite. Because each method applies to a different case, choose the method that most resembles your situation. The three methods are listed below and are explained briefly.

### **Session 4: Loading from a Device**

In this Session, you load device data from a previously programmed device (a master device). If you do not have a master device and you still want to follow the steps in the Session, you can use a blank device. (Although the data loaded into UniSite will be blank, you will have a chance to follow the procedures.)

### **Session 5: Loading Data from Disk**

In this Session, you load device data from UniSite's internal disk drive. A sample data file has been provided on the UniSite System disk for you to use.

### **Sessions 6, 7, & 8: Loading Data from a Port**

In these Sessions, you load device data into UniSite from a host, such as a PC or workstation connected to one of the serial ports on UniSite. If you are using a PC, you need a terminal emulation package (like Data I/O's HiTerm). If you are using a workstation, you need to know how to transfer a file out of the workstation through the workstation's serial port.

## Session 4: Loading Data from a Device

This Session illustrates how to load data into UniSite from a master device. (A master device is a device that is already programmed.) The device used for this session is an AMD 27256 DIP EPROM. If you have a different memory device, you can still follow along, substituting your device manufacturer and part number in place of the AMD 27256.

If you do not have a master device, we suggest you go to the next Session, which shows you how to load data from UniSite's disk drive. When you have completed Session 4, skip the next three Sessions and continue with Session 9.

### Before You Begin

Before starting this Session, you should have completed the first two Sessions. Specifically, you should be familiar with UniSite's interface, and you should know how to select a device.

Also, you should have Site48 (or Site40) installed in UniSite. If Site48 (or Site40) is not installed in UniSite, or if you would like to review the instructions for installing/removing Site40 or Site48, see Chapter 2, "Setup and Installation." Read the section titled "Installing a Module" for more information.

In any event, you should not go any further until you are familiar with installing/removing a module or inserting/removing a device.

### Can I Use Another Device?

If you do not have an AMD 27256 (the device we are going to use for this Session), then you should go to the Select Device screen and select the device you are going to program. Keep in mind that the device you select might not have the same capabilities as the AMD 27256. For example, the AMD 27256 supports Electronic ID while the Hitachi 27256 does not.

### Select a Device

Before loading data from a device, you must first tell UniSite what type of device you are using.

Choose Select Device from the Main Menu. Select the manufacturer and part number of the master device.

Remember, if you don't see the part number on the first screen of part numbers, press **Ctrl** + **N** to display the next screen of part numbers. Also, if you don't see the device part number you are looking for, go back to the Manufacturer List and look at the Device Type field. This field allows you to filter out certain device types. Press **Space** to cycle through the three settings: All, Memory & Emicros, or Logic Only. When you select a manufacturer, UniSite displays only devices that fit the filter you selected.

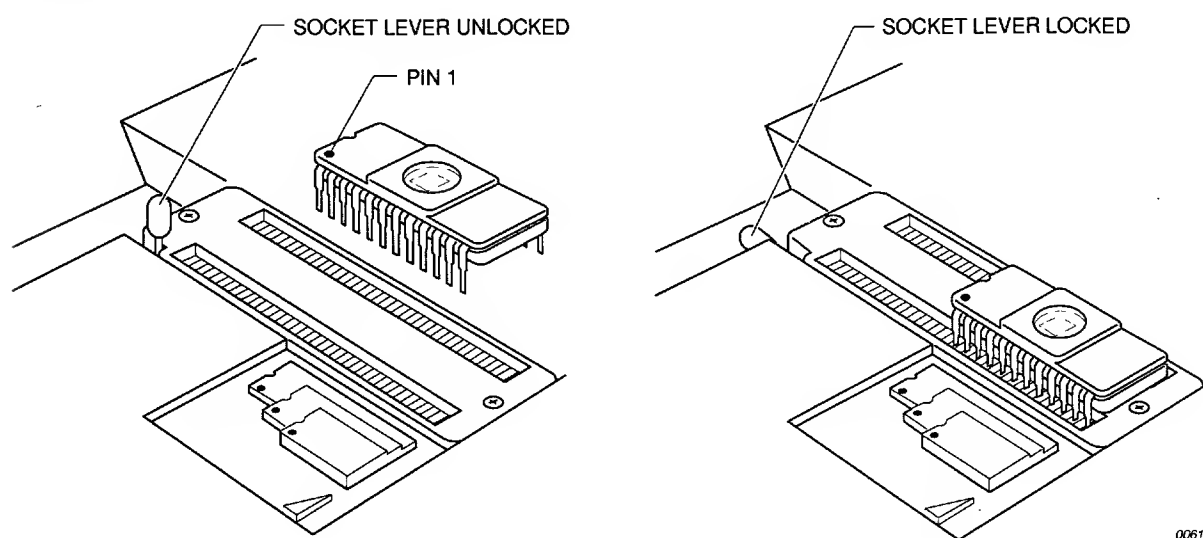
When the programming algorithm has finished loading, the status window will display the selected device and you will return to the Main Menu. Also, UniSite will set the load parameters to match the size of the selected device.

## Insert the Device

Make sure Site48 is properly installed in UniSite. If the device socket is locked, unlock it by pulling up the socket lever. Insert the device into the device socket, making sure that the device is bottom-justified and that pin 1 is in the upper-left corner.

Lock the device into place by pressing the socket lever down. Figure 4-8 shows examples of locked and unlocked sockets and also shows the proper orientation for a DIP device.

**Figure 4-8**  
*Inserting a DIP Device*



0061-2

*Note: Insert DIP devices into UniSite AFTER you have a module installed in UniSite.*

## Load the Data

With the device type selected and the master device in the socket, the next step is to load the data into UniSite's RAM. (You could also store the data on the disk drive, but RAM is faster.)

Choose Load Device from the Main Menu. The Load Memory Device parameter screen, shown in Figure 4-9, is displayed.

**Figure 4-9**  
*The Load Memory Device  
Parameter Screen*

FILENAME:		RAM AVAIL: 128 OF 128KB		REV: X.XX Y.YY Z.Z	
MANUFACTURER: AMD		PART #: Z7Z56		FAMILY/PIN CODE: 0C1 / 032	
I/O FORMAT:					
MAIN MENU					
Select device	LOAD MEMORY DEVICE (non-default)				
Quick copy	Destination (RAM, Disk) <input type="checkbox"/>				
Load device	Data word width <input type="checkbox"/>				
Program device	Next device <input type="checkbox"/>				
Verify device	Total set size <input type="checkbox"/>				
More commands	User data size <input type="checkbox"/>				
	Next operation begins at <input type="checkbox"/>				
Return: Execute		F4: Select mode/options		F3 or ? : Help	
F1: Main menu		F2: Prev menu			

## About Parameters

Parameters are user-definable fields that either qualify or quantify UniSite's actions. Qualifying parameters, such as Illegal Bit Check, control whether or not UniSite performs an operation. Quantifying parameters, such as Block Size or I/O Translation Format, give UniSite a range or variable to use in an operation.

Six different parameters are shown in Figure 4-9, including Destination and User Data Size.

## Parameter Screens

Look at the Load Memory Device screen. At the top of the dialog window, you see one of the following:

LOAD MEMORY DEVICE (all parameters)

LOAD MEMORY DEVICE (non-default)

There are two types of parameter screens: simple and complex. UniSite defaults to displaying the simple, Non-default parameters screen.

## What Is the Difference?

Both screens let you change load parameters. The All Parameters screen contains all the load parameters supported by the selected device, while the Non-default parameters screen contains a simplified subset of the load parameters supported by the selected device.

To see the difference between the two screens, press **F4**. If you were looking at the Non-default screen, you are now be looking at the All Parameters screen. (Likewise, if you were looking at the All Parameters screen you are now be looking at the Non-default screen.) As you can see, there is roughly a 2:1 difference in the number of parameters between the two screens. The All Parameters screen is shown in Figure 4-10. Compare the difference between the two screens.

**Figure 4-10**  
*The All Parameters Load Memory Screen*

FILENAME:		RAM AVAIL: 128 OF 128KB		REV: X.XX Y.YY Z.Z	
MANUFACTURER: AMD		PART #: 27256		FAMILY/PIN CODE: 0C1 / 032	
I/O FORMAT:					
<div> <div> MAIN MENU Select device Quick copy <b>Load device</b> Program device Verify device More commands </div> <div> LOAD MEMORY DEVICE (all parameters) </div> </div>					
Destination (RAM, Disk)		<input type="checkbox"/> RAM <input checked="" type="checkbox"/> Disk			
Data word width		<input type="text" value="8"/>			
Next device		<input type="text" value="1"/>			
Total set size		<input type="text" value="1"/>			
User data size		<input type="text" value="8000"/>			
Next operation begins at		<input type="text" value="0"/>			
Memory begin address		<input type="text" value="0"/>			
Device begin address		<input type="text" value="0"/>			
Device block size		<input type="text" value="8000"/>			
Set auto-increment		<input checked="" type="checkbox"/>			
Compare elec ID		<input checked="" type="checkbox"/>			
Return: Execute      F4: Select mode/options F1: Main menu      F2: Prev menu      F3 or 7: Help					

**Which Should I Use?** During the course of normal device operations, you will usually need only the parameters on the Non-default screen. But if you want to do more complicated operations, you need to change some of the parameters on the All Parameters screen. A complicated load operation might be loading only part of the data from a device.

You will not be doing any complicated load operations for this Session, so you should be looking at the Non-default screen. If the All Parameters screen is displayed, press **[F4]** to switch to the Non-default screen.

**Setting the Parameters** Look again at the parameters on the screen. With these parameters, you can change the destination, word width, memory begin address, and block size of the data to be loaded. But, for this Session, make sure that the parameters are set to the values shown in Figure 4-9.

**Changing a Parameter** To change the value of a parameter entry, move the cursor to the desired field and type the new value. Then either press **[J]** or use the arrow keys to move the cursor to a new field. If you try to enter an incorrect parameter, UniSite beeps and the message line reads

Illegal parameter value

If you enter a valid parameter, the message line reads

Parameter Entered

For more information on any of the parameters, see Chapter 5, "Commands." (Or you could use online help to get information on each of the parameters on the Load Memory Device screen.)

**Loading the Data** Now that you have set the parameters, you are ready to load the data from the device.

To begin the load, press **[J]**. When UniSite has loaded the data, it displays the following message in the message bar:

OPERATION COMPLETE: Sumcheck = xxxxxxxx

where xxxxxxxx represents the 8-digit sumcheck of the data loaded from the device.

**Review** When you select Load Device from the Main Menu, you see one of two parameter-entry screens: the simple Non-default screen or the complex All Parameters screen. Press **[F4]** to toggle between the two parameter-entry screens.

When you have entered the parameters, press **[J]** to begin the loading.

When UniSite is finished loading the data from the device, it displays an 8-digit sumcheck of the data loaded. (UniSite displays a 4-digit sumcheck when you load data from a logic device.)



---

## Session 5: Loading Data from Disk

This Session illustrates how to load binary device data into UniSite from one of UniSite's internal disk drives. To help you with this Session, we have included a data file on UniSite System disk. You can use the sample data file to practice loading data files from a disk in one of UniSite's disk drives. The sample file is named **sample.bin**.

When you have completed this Session, skip the next two Sessions and continue with Session 8. If you have a data file you want to download to UniSite, you should skip this Session and complete the next two Sessions.

### Before You Begin

Before starting this Session, you should have completed the first two Sessions. Specifically, you should be familiar with UniSite's user interface, and you should know how to select a device.

### Getting There

Go to the More Commands menu and notice the following two menu items:

File Operations

Transfer Data

They both sound like logical places to put a command for loading a data file. In fact both menus contain commands for loading data files, but the two commands manipulate data in different ways.

### What's the Difference?

The Transfer Data command calls up the Transfer Menu. On this menu, you will find commands used for transferring formatted data files between UniSite and another machine (like a PC). You will also find commands for transferring **formatted** data files (not binary files) between UniSite and a disk in one of UniSite's disk drives. This menu does not contain the command you are looking for.

The File Operations command calls up the File Menu. On this menu you will find DOS-like commands: such as load, rename, delete, format disk, and copy disk. The command you are looking for, the Load Data command, is on the File Menu. The Load Data command simply loads the binary data file directly into memory.

### When Would I Use this Command?

Use the Load Data command if the data file you want to load contains a binary image of the data you want to program into a device. The Save File command creates a binary image file by copying the data in UniSite's RAM directly to a disk file.

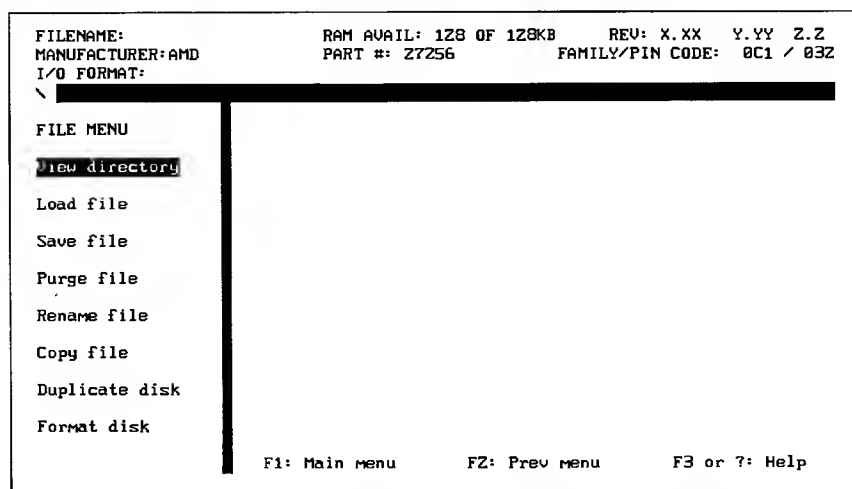
---

*Note: Do not use the Load Data and Save File commands with formatted data files. Use the Input from Disk and Output to Disk commands to load and save formatted data files. See Chapter 5, "Commands," for more information.*

## The File Menu

To get to the File menu, select File Operations from the More Commands menu. The File menu appears and should look like the screen shown in Figure 4-11.

**Figure 4-11**  
The File Menu



## Viewing a Directory

The View Directory command displays a directory of the files stored on the disks in both disk drives. The files on the disk in the lefthand drive (drive A:) are prefixed with A: and the files on the disk in righthand drive (drive B:) are prefixed with B:. The sample data file is named **sample.bin**. Select the View Directory command and locate the test file.

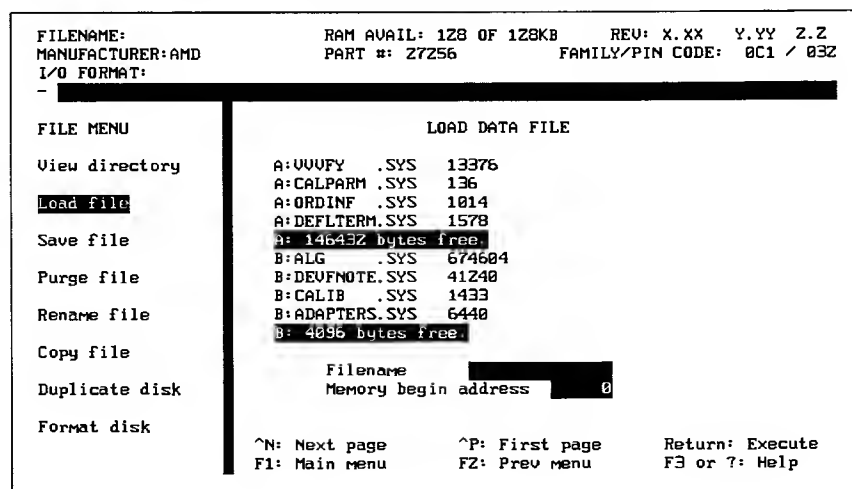
If you don't see **sample.bin**, don't worry; UniSite displays only 28 files at one time. Press **Ctrl** + **N** to see the next page of files.

When you have found **sample.bin**, press **F2** to return to the File menu.

## Loading the File

Now that you know the name of the file (**sample.bin**), and have found the disk it is on, you are ready to load the file. Select Load File from the File menu. The dialog window displays a directory of the files on the disks in UniSite's two disk drives. (Notice that the Load File command displays a directory similar to the View Directory command.) A screen similar to what you will see is shown in Figure 4-12.

**Figure 4-12**  
The Load File Dialog Screen



To load the sample data file, follow the steps below:

1. Move the cursor to the Filename field and type *b:sample.bin* .  
UniSite displays the following message in the message bar:
2. Next, move the cursor to the Memory Begin Address field and enter 0 .
3. Press  to begin loading the data file. The action symbol rotates while UniSite loads the data file, and displays the following message in the message bar:

Loading data from file.

If the data file loads successfully, UniSite displays the following message in the message bar:

Done.

### Did the File Load Successfully?

If the sample data file did not load successfully, return to the beginning of this Session and try it again.

If the sample data file loaded successfully, you will be returned to the Load File dialog screen. Skip to Session 8. In that Session you will find out what the sample file contains, and you will learn how to edit the sample file.

## Review

You can load device data into UniSite by loading a data file from one of UniSite's internal disk drives. Use the commands on the File Operations menu (such as the Load File command) if your data file is stored in binary image format. Use the commands on the Transfer Data menu (such as the Input from Disk command) if your data file is stored in a specific data translation format, such as Intel Hex or JEDEC. There is a list of supported formats at the beginning of Chapter 7, "Translation Formats."

When you select the Load File command, UniSite displays a directory of the files on the disks in UniSite's two drives. Enter the drive name and filename of the file you want to load. For example, enter *a:sample.bin*.

If you do not see the file you want to load, press Ctrl + N to display the next page of files, or press Ctrl + P to display the previous page of files. If you are loading data for a memory device, you should also enter a memory begin address.

Finally, press  to begin the loading. During the load operation, the action symbol rotates. If the load operation completes successfully, UniSite displays the following message in the message bar:

Done.

## Session 6: Selecting a Translation Format

This Session and the following Session are companion Sessions. This Session introduces you to translation formats and shows you how to select a translation format. The next Session shows you how to load a data file through UniSite's Terminal port.

### About Translation Formats

Translation formats represent different ways of representing the device data in a data file. A data file could contain the fuse pattern and test vectors for a logic device or the data for a memory device.

### Getting There

Starting at the Main Menu, press **[M][T][F]** to get to the Translation Format screen, which should look like Figure 4-13. If you get lost, return to the Main Menu and start over. (Remember, press **[F1]** to get to the Main Menu.)

**Figure 4-13**  
The Translation Format  
Selection Screen

```

- I/O TRANSLATION FORMAT

09 5 Level BNPF W/O STX      36 ASCII-Oct "/" With SOH  92 JEDEC (kernel)
08 5 Level BNPF With STX     31 ASCII-Oct "/" With STX  81 MOS Technology
14 Altera POF                32 ASCII-Oct Apostrophe    82 Motorola Exorcise
07 ASCII-B10F W/O STX        37 ASCII-Oct SMS           87 Motorola Exoramax
03 ASCII-B10F With STX       35 ASCII-Oct SP With SOH   95 Motorola S3
06 ASCII-BH1F W/O STX        30 ASCII-Oct SP With STX   70 RCA Cosmac
02 ASCII-BH1F With STX       10 Binary                  85 Sig Absolute Obj
05 ASCII-BNPF W/O STX        11 DEC Binary              13 Spectrum W/O STX
01 ASCII-BNPF With STX       80 Fairchild Fairbug       12 Spectrum With STX
56 ASCII-Hex "/" With SOH    89 HP 64000 Absolute       94 Tek Hex Extended
51 ASCII-Hex "/" With STX    96 HP Unix                 86 Tek Hex
58 ASCII-Hex "." With SOH    99 Intel Hex-32            90 TI SDSMAC
53 ASCII-Hex "." With STX    83 Intel Intellec 8/MDS    04 TI SDSMAC (320)
52 ASCII-Hex Apostrophe     88 Intel MCS-86 Hex Obj
57 ASCII-Hex SMS            98 Intel OMF286
55 ASCII-Hex SP With SOH    97 Intel OMF386
50 ASCII-Hex SP With STX    91 JEDEC (full)

Select translation format 00

F1: Main menu      F2: Prev menu      F3 or ?: Help      Return: Execute
  
```

### Making a Choice

Now that you are at the I/O Translation Format selection screen, you need to choose a translation format to use. Normally, you would select the same format as your data file. For this Session, select the Intel 8-bit Hex translation format.

Select the translation format the same way you selected a device manufacturer: find what you are looking for and type the number beside it. So, for the format selection screen shown in Figure 4-13, you would type **[8][3][ ]** to select the Intel 8-bit Hex (Intel Intellec 8/MDS) translation format. When you press **[ ]** UniSite configures itself for the selected translation format and returns you to the Transfer Menu. Notice that the I/O Format line in the status window has changed and now reads

```
I/O FORMAT: Intel Intellec 8/MDS
```

### Review

Use the Format Select command on the Transfer Data menu to select a new translation format. After locating the desired format, type the number to the left of the format.

---

## Session 7: Loading Data from a PC Using HiTerm

In this Session, you will learn how to use HiTerm to download data from a PC to UniSite through one of the serial ports on UniSite.

---

*Note: If you are using a DOS-based PC, we recommend you use HiTerm as your terminal emulation software. Although you might already have other terminal emulation software on your PC, HiTerm was written to work with UniSite and can download files to UniSite at 115.2K baud.*

### Why HiTerm?

At this point, you might be wondering why we recommend you use HiTerm and not another terminal emulator. First and foremost, only HiTerm supports 115.2K baud downloading. Second, HiTerm supports formatted binary transfer, while some other terminal emulators do not. Third, HiTerm was written by Data I/O with UniSite in mind. HiTerm does all the opening and closing of files automatically; with other terminal emulators you must do that yourself.

### Before You Begin

This Session assumes you have a DOS-based PC connected to UniSite. See Chapter 2, "Setup and Installation," for information on connecting a PC to UniSite.

Also, you must have HiTerm properly installed on the PC connected to UniSite. See Chapter 2, "Setup and Installation," for quick installation instructions for HiTerm. See the *HiTerm User Manual* behind the Utilities tab in this binder if you need more information about installing HiTerm.

Finally, we suggest you have UniSite configured for High Speed Download, which allows you to download data file to UniSite at 115.2K baud. See the "Setting Up High Speed Download" section at the end of Chapter 2 for more information about configuring UniSite for High Speed Download.

### What to Transfer?

We suggest you use **sample.dat**, a sample data file supplied on the HiTerm disk. If you want to transfer a different data file, substitute its filename for **sample.dat**. If your data file is in a format other than Intel Intellec 8/MDS, use the Format Select command to select that format. See Session 6 for information on selecting a translation format.

## Preparing UniSite

**Figure 4-14**  
The Download Data from Host Screen

From the Main Menu, press **[M][T][D]** to get to the Download Data from Host screen, which is shown in Figure 4-14.

FILENAME:                      RAM AVAIL: 128 OF 128KB      REV: X.XX Y.YY 2.2 MANUFACTURER: AMD            PART #: 27256            FAMILY/PIN CODE: 8C1 / 832 I/O FORMAT: Intel Intellec 8/MDS	
<b>TRANSFER MENU</b> Download data Upload data Compare data Format select Input from disk Output to disk Serial output	<b>DOWNLOAD DATA FROM HOST</b> Source (Remote, Terminal)      R Destination (RAM, Disk)        R  I/O Translation Format            83 I/O addr offset                  FFFFFFFF Memory begin address            0 User data size                    0  Download host command  F1: Main menu      F2: Prev menu      F3 or ? : Help

Look at the parameters and make sure they reflect your system configuration. If you are running HiTerm on a PC connected to the Remote port on UniSite, use the following settings:

- Source – Remote port
- Destination – RAM
- I/O Translation Format – 83 (Intel Hex)
- I/O Address Offset – FFFFFFFF
- Memory Begin Address – 0
- User Data Size – 0

If your configuration is different, change the parameters to match your configuration. For example, select translation format 91 if the data file you will download is a JEDEC file.

For now, leave the Download Host Command blank; you will fill that in later. If the parameters are correct, skip ahead to the section titled "Downloading the File."

*Note: See Chapter 5, "Commands," if you want to find out more about the parameters on the Download Data from Host screen.*

## About Parameters

Parameters are user-definable fields that determine what UniSite does. Parameters either qualify or quantify UniSite's actions. Qualifying parameters, such as Source and Destination, control the type of operation to perform. Quantifying parameters, such as Block Size or I/O Translation Format, give UniSite a range or variable to use in an operation.

Seven parameters are shown in Figure 4-14, including Source, Memory Begin Address, and Download Host Command.

### Changing a Parameter

Follow the steps below if you need to change a parameter.

1. Move the cursor to the field you need to change.
2. Type the new value and move the cursor to another field. If the parameter is acceptable, UniSite displays

Parameter Entered

in the message bar. (Remember, the message bar is located below the status window.)

If you enter an incorrect parameter, UniSite beeps and displays an error message. Continue until the displayed parameters reflect your configuration.

### Downloading the File

#### Locate the Data File on the PC

Change to the directory containing the data file you are going to download. (Remember, if you are using the sample data file supplied with HiTerm, you are looking for the **sample.dat** file.)

If you are in DOS, use the CD and DIR commands to change to the directory containing the data file to download.

If you are running HiTerm, press **[Alt] + [F6]** **[J]** to view the current directory. If **sample.dat** is in the current directory, press **[J]** to return to the terminal emulation.

---

*Note: The function key commands in this Session are for IBM-compatible PCs. However, if you are using an NEC-9800 PC, the HiTerm commands will be slightly different. Consult the HiTerm documentation for more information.*


If **sample.dat** is not in the current directory, press **[Alt] + [F5]** to bring up HiTerm's change directory command. Type the drive and pathname of the directory containing **sample.dat** and press **[J]** to change to that directory. Press **[Alt] + [F6]** **[J]** to view the current directory. If **sample.dat** is in the current directory, press **[J]** to return to the terminal emulation. If the data file is not in the current directory, repeat the procedure until you find the data file you want to download.

#### About the Download Host Command

Earlier in this Session, you entered the parameters for the download, but you left the Download Host Command parameter blank. The Download Host Command is a command that UniSite sends to the PC to initiate the download. Because you are running HiTerm, you can specify a special Download Host Command, the TRANSFER command.

When UniSite sends the TRANSFER command (or TR for short) it tells HiTerm to download a specified file to UniSite.


If the sample data file is in your current directory, move the cursor to the Download Host Command field and enter the following Download Host Command

`tr sample.dat` 

Remember to substitute the appropriate filename if you are using a different file. When you enter the Download Host command, the message bar displays

Parameter Entered

### Download the File

Press  to begin the download. The action symbol rotates while the data is being downloaded. When the download is finished the message bar displays

Data transfer complete. Data sum = xxxxxxxx

Go to the section titled "After the Download" when the download is complete.

### After the Download

By this point, you should have successfully downloaded a data file into UniSite.

### About the Sumcheck

When the download was completed, the message bar displayed

Data transfer complete. Data sum = xxxxxxxx




This message tells you two things: first, it tells you that the data transfer completed successfully, and second, it tells you what the sumcheck of the data transferred is. The sumcheck is shown above as xxxxxxxx.

The sumcheck for memory devices is an eight-digit hexadecimal summation of the data downloaded. If you change one byte of information in the data file, the sumcheck will also change. The sumcheck is a good method of verifying that the data you downloaded matches the data on your host.

Later, when you program this data into a device, UniSite will generate another sumcheck. If the two are the same, then the data programmed is the same as the data downloaded. If the two sumchecks are different, then the data programmed is not the same as the data downloaded.

### Review

In this Session, you learned how to download data to UniSite from a host connected to UniSite's Terminal port. The steps were

- Select the translation format that matched the format of your data file.
- Go to the download screen by pressing   .
- Set the parameters on the Download screen to match your setup.
- Enter the host Download Host Command. For example, to transfer the file `filename.dat` you would type `tr filename.dat` for a PC using HiTerm.



## Session 8: Loading Data from a Host

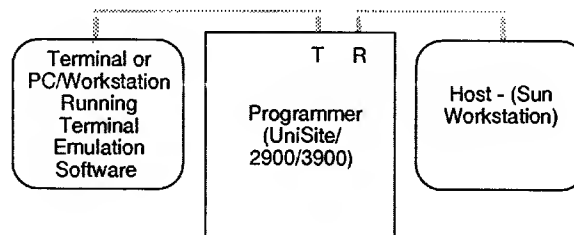
In this Session, you will learn how to download data from a host to UniSite through one of the serial ports on UniSite. The procedures in this Session apply to many types of hosts, including VAXes, UNIX-based workstations, and DOS-based PCs.

*Note: If you are using a DOS-based PC, we recommend that you use HiTerm as your terminal emulation software. Downloading a file with HiTerm is covered in the previous Session.*

### Before You Begin

This Session assumes that you have a host connected to one of the serial ports on UniSite and a terminal connected to the other serial port on UniSite. This type of configuration is called Transparent mode and is shown in Figure 4-15.

**Figure 4-15**  
Transparent Mode



0999-2

See Chapter 2, "Setup and Installation," for information on connecting a host and terminal to UniSite.

### About Transparent Mode

Transparent mode, shown in Figure 4-15, is a feature of UniSite that allows the programmer to be inline between your terminal and host computer. This eliminates the need for a switch box or a second link to the host, and enables you to download directly from the host to UniSite. The host could be a networked file server such as a VAX or a Sun. When setup properly, the terminal connected to UniSite can control both UniSite and the remote host.

In Transparent mode, UniSite passes all characters through its Terminal and Remote ports as if it weren't there. The two serial ports on UniSite can even operate at different baud rates. While operating UniSite from the terminal, press **[Esc] [Ctrl] + [T]** to toggle UniSite between terminal mode and transparent mode. UniSite remains in transparent mode until it receives another **[Esc] [Ctrl] + [T]** command, at which time it switches back to terminal mode.

### What to Transfer?

Using your development tools, create a small sample data file. For the rest of this Session, the sample data file will be referred to as **sample.dat**. If you give your sample data file a different name, substitute that name where you see **sample.dat**.

Also, the rest of this Session assumes that your sample data file is stored in the Intel Intellec 8/MDS data translation format. If your data file is in a different translation format, use the Format Select command to select that format. See Session 6 for information on selecting a data translation format.

## Preparing UniSite

**Figure 4-16**  
*The Download Data from Host Screen*

From the Main Menu, press **[M][T][D]** to get to the Download Data From Host screen, which is shown in Figure 4-16.

FILENAME:		RAM AVAIL: 128 OF 128KB		REV: X.XX Y.YY Z.Z	
MANUFACTURER: AMD		PART #: Z7Z56		FAMILY/PIN CODE: 0C1 / 03Z	
I/O FORMAT: Intel Intellec 8/MDS					
/					
TRANSFER MENU		DOWNLOAD DATA FROM HOST			
Download data		Source (Remote, Terminal)		R	
Upload data		Destination (RAM, Disk)		R	
Compare data		I/O Translation Format		83	
Format select		I/O addr offset		FFFFFFFF	
Input from disk		Memory begin address		0	
Output to disk		User data size		0	
Serial output		Download host command			
		F1: Main menu F2: Prev menu F3 or ? : Help			

Look at the parameters and make sure they reflect your system configuration. Use the following settings if your host is connected to the Remote port on UniSite:

- Source – Remote port
- Destination – RAM
- I/O Translation Format – 83 (Intel Hex)
- I/O Address Offset – FFFFFFFF
- Memory Begin Address – 0
- User Data Size – 0

If your configuration is different, change the parameters to match your configuration. For example, select translation format 91 if the data file you will download is a JEDEC file.

For now, leave the Download Host Command blank; you will fill that in later. If the parameters are correct, skip ahead to the section titled "Downloading the File."

## About Parameters

Parameters are user-definable fields that determine what UniSite does. Parameters either qualify or quantify UniSite's actions. Qualifying parameters, such as Source and Destination, control the type of operation to perform. Quantifying parameters, such as Block Size or I/O Translation Format, give UniSite a range or variable to use in an operation.

Seven parameters are shown in Figure 4-16, including Source, Memory Begin Address, and Download Host Command.

### Changing a Parameter

Follow the steps below if you need to change a parameter:

1. Move the cursor to the field you need to change.
2. Type the new value and move the cursor to another field. If the parameter is acceptable, UniSite displays

Parameter Entered

in the message bar. (Remember, the message bar is located below the status window.)

If you enter an incorrect parameter, UniSite beeps and displays an error message. Continue until the displayed parameters reflect your configuration.

## Downloading the File

### Locate the Data File on the Host

Change to the directory containing the file you are going to download. Before you change directories on the host, press **[Esc] [Ctrl] + [T]** to enter Transparent mode.

If you are using a UNIX-based host, use the CD command to change to the directory containing the data file to download.

---

*Note: The commands in this Session are for UNIX-based hosts. Substitute the appropriate commands if you are using a different type of host, such as a VMS-based machine. If you need more information, consult the system's documentation or your system administrator.*

Once you have changed to the directory containing the data file you are going to download, press **[Esc] [Ctrl] + [T]** again to leave Transparent mode and return to UniSite's full screen interface. You might have to press **[Ctrl] + [R]** to redraw the screen.

### About the Download Host Command

Earlier in this Session, you entered the parameters for the download, but you left the Download Host Command parameter blank. The Download Host Command is a command that UniSite sends to the host to initiate the download. Because you are on a UNIX-based host, use the CAT command as the Download Host Command.

If the sample data file is in your current directory, enter the following Download Host Command

```
cat sample.dat 
```

Remember to substitute the appropriate filename if you are using a different file. When you enter the Download Host command, the message bar displays

Parameter Entered

### Download the File

Press **[J]** to begin the download. The action symbol rotates while the data is being downloaded. When the download is finished the message bar displays

```
Data transfer complete. Data sum = xxxxxxxx
```

### After the Download

By this point, you should have successfully downloaded a data file into UniSite.

### About the Sumcheck

When the download was completed, the message bar displayed

```
Data transfer complete. Data sum = xxxxxxxx
```

This message tells you two things: first, it tells you that the data transfer completed successfully, and second, it tells you what the sumcheck of the data transferred is. The sumcheck is shown above as *xxxxxxx*.

The sumcheck for memory devices is an eight-digit hexadecimal summation of the data downloaded. If you change one byte of information in the data file, the sumcheck will also change. The sumcheck is a good method of verifying that the data you downloaded matches the data on your host.

Later, when you program this data into a device, UniSite will generate another sumcheck. If the two are the same, then the data programmed is the same as the data downloaded. If the two sumchecks are different, then the data programmed is not the same as the data downloaded.

### Review

In this Session, you learned how to download data to UniSite from a host connected to the Remote port on UniSite. The steps were

- Select the translation format that matched the format of your data file.
- From the Main Menu, press **[M][T][D]** to go to the download screen.
- Set the parameters on the Download screen to match your setup.
- Enter the host Download Host Command. For example, to transfer the file **filename.dat** you would enter **cat filename.dat** for a UNIX-based host. If you are using a VMS-based host, you would enter **type filename.dat**.

## Session 9: Editing Data

In the previous Session, you loaded a data file into UniSite. In this Session, you will learn how to edit that data file.

### Before You Begin

Return to the Main Menu before you begin this Session. Also, you should select a memory device before you begin this Session. See Session 2 for a quick explanation of how to select a device.

### Getting There

The Edit Data command is located on the More Commands menu. Select More Commands/Edit Data from the Main Menu. Finally, select Edit Memory to bring up the Edit Programmer Memory dialog window, which is shown in Figure 4-17.

**Figure 4-17**  
The Edit Programmer  
Memory Screen

FILENAME:                      RAM AVAIL: 128 OF 128KB                      REV: X.XX   Y.YY   Z.Z MANUFACTURER: AMD                      PART #: 27256                      FAMILY/PIN CODE: 0C1 / 032 I/O FORMAT: Intel Intellec 8/MDS	
<b>EDIT MENU</b> Edit memory Complement Move data Fill memory	<b>EDIT PROGRAMMER MEMORY</b> Source (RAM, Disk) <input type="checkbox"/> <input checked="" type="checkbox"/> Edit data word width (4, 8, 16) <input checked="" type="checkbox"/> 8 Edit address offset <input type="text" value="0"/> Edit begin address <input type="text" value=""/> Return: Execute F1: Main menu                      F2: Prev menu                      F3 or ?: Help

Make sure the parameters are set as they are shown in Figure 4-17, then press ☐. (For an explanation of how to change parameters, refer to the "Changing a Parameter" section of Session 7.) The entire screen clears and the edit screen appears, as shown in Figure 4-18.

**Figure 4-18**  
The Edit Screen

CURSOR AT LOCATION: 00000094      8 BIT ADDRESSING																																																																																																																																																																																																																																																																																																																													
<table border="0"> <tr> <th>ADDRESS</th> <th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th> <th>HEXADECEMAL</th> <th>ASCII</th> </tr> <tr> <td>00000000</td> <td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td> <td>0123456789ABCDEF</td> </tr> <tr> <td>00000010</td> <td>54</td><td>68</td><td>69</td><td>73</td><td>20</td><td>69</td><td>73</td><td>20</td><td>61</td><td>6E</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td> <td rowspan="4">This is an example of some programming data that has been loaded into</td> </tr> <tr> <td>00000020</td> <td>65</td><td>78</td><td>61</td><td>6D</td><td>70</td><td>6C</td><td>65</td><td>20</td><td>6F</td><td>66</td><td>20</td><td>73</td><td>6F</td><td>6D</td><td>65</td><td>20</td> </tr> <tr> <td>00000030</td> <td>70</td><td>72</td><td>72</td><td>67</td><td>72</td><td>61</td><td>6D</td><td>6D</td><td>69</td><td>6E</td><td>67</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td> </tr> <tr> <td>00000040</td> <td>64</td><td>61</td><td>74</td><td>65</td><td>20</td><td>74</td><td>68</td><td>61</td><td>74</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td> </tr> <tr> <td>00000050</td> <td>68</td><td>61</td><td>73</td><td>20</td><td>62</td><td>65</td><td>65</td><td>6E</td><td>20</td><td>6C</td><td>6F</td><td>61</td><td>64</td><td>65</td><td>64</td><td>20</td> <td rowspan="4">..... ..... .....</td> </tr> <tr> <td>00000060</td> <td>69</td><td>6E</td><td>74</td><td>6F</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td> </tr> <tr> <td>00000070</td> <td>2E</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td> </tr> <tr> <td>00000080</td> <td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td><td>20</td> </tr> <tr> <td>00000090</td> <td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td> <td rowspan="4">..... ..... .....</td> </tr> <tr> <td>000000A0</td> <td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td> </tr> <tr> <td>000000B0</td> <td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td> </tr> <tr> <td>000000C0</td> <td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td> </tr> <tr> <td>000000D0</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr> <td>000000E0</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr> <td>000000F0</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr> <td colspan="18">         ^P: Prev block                      ^E: Exchange                      with                      ^U: Restore block          ^N: Next block                      ^F: Search pattern                      ^B: Jump to address                      F2: Exit editor          ^D: Delete byte                      insert                      Tab: Toggle Hex/ASCII modes          ^T: Start/stop       </td> </tr> </table>		ADDRESS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	HEXADECEMAL	ASCII	00000000	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	0123456789ABCDEF	00000010	54	68	69	73	20	69	73	20	61	6E	20	20	20	20	20	20	This is an example of some programming data that has been loaded into	00000020	65	78	61	6D	70	6C	65	20	6F	66	20	73	6F	6D	65	20	00000030	70	72	72	67	72	61	6D	6D	69	6E	67	20	20	20	20	20	00000040	64	61	74	65	20	74	68	61	74	20	20	20	20	20	20	20	00000050	68	61	73	20	62	65	65	6E	20	6C	6F	61	64	65	64	20	..... ..... .....	00000060	69	6E	74	6F	20	20	20	20	20	20	20	20	20	20	20	20	00000070	2E	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00000080	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	..... ..... .....	000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	000000D0																		000000E0																		000000F0																		^P: Prev block                      ^E: Exchange                      with                      ^U: Restore block ^N: Next block                      ^F: Search pattern                      ^B: Jump to address                      F2: Exit editor ^D: Delete byte                      insert                      Tab: Toggle Hex/ASCII modes ^T: Start/stop																	
ADDRESS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	HEXADECEMAL	ASCII																																																																																																																																																																																																																																																																																																											
00000000	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	0123456789ABCDEF																																																																																																																																																																																																																																																																																																												
00000010	54	68	69	73	20	69	73	20	61	6E	20	20	20	20	20	20	This is an example of some programming data that has been loaded into																																																																																																																																																																																																																																																																																																												
00000020	65	78	61	6D	70	6C	65	20	6F	66	20	73	6F	6D	65	20																																																																																																																																																																																																																																																																																																													
00000030	70	72	72	67	72	61	6D	6D	69	6E	67	20	20	20	20	20																																																																																																																																																																																																																																																																																																													
00000040	64	61	74	65	20	74	68	61	74	20	20	20	20	20	20	20																																																																																																																																																																																																																																																																																																													
00000050	68	61	73	20	62	65	65	6E	20	6C	6F	61	64	65	64	20	..... ..... .....																																																																																																																																																																																																																																																																																																												
00000060	69	6E	74	6F	20	20	20	20	20	20	20	20	20	20	20	20																																																																																																																																																																																																																																																																																																													
00000070	2E	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20																																																																																																																																																																																																																																																																																																													
00000080	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20																																																																																																																																																																																																																																																																																																													
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	..... ..... .....																																																																																																																																																																																																																																																																																																												
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																																																																																																																																																																																																																																																																																																													
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																																																																																																																																																																																																																																																																																																													
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00																																																																																																																																																																																																																																																																																																													
000000D0																																																																																																																																																																																																																																																																																																																													
000000E0																																																																																																																																																																																																																																																																																																																													
000000F0																																																																																																																																																																																																																																																																																																																													
^P: Prev block                      ^E: Exchange                      with                      ^U: Restore block ^N: Next block                      ^F: Search pattern                      ^B: Jump to address                      F2: Exit editor ^D: Delete byte                      insert                      Tab: Toggle Hex/ASCII modes ^T: Start/stop																																																																																																																																																																																																																																																																																																																													

### What Does It Say?

The edit screen displays 256 bytes at a time. The four-line reminder bar at the bottom of the screen displays the available commands (no online help is available for this screen). The right third of the screen displays the ASCII translation of the hexadecimal bytes on the left portion of the screen. The top line of the screen displays the address location of the cursor (in hexadecimal) and the addressing mode.

If you haven't already, read the message in the sample data file. As you can see there are some errors that need correcting.

### Moving Around

The cursor starts in the upper-left corner of the hex display. Move the cursor around and watch what happens to the cursor location counter (on the top line of the screen). As you move the cursor, the counter changes to reflect your current location. If you wanted to edit hex data, you would do it in the hex column. But during this Session, you will be editing text, so you should move the cursor to the ASCII side of the screen.

To move to the ASCII side of the screen, press **Tab**. Move the cursor around. Notice that the cursor location counter still changes when you move from one address to another.

### Making Changes

#### Overtyping

With the cursor in the ASCII field, move the cursor to the second **r** in **prrogramming**. Type **o** to correct the spelling mistake. Next, move the cursor to **date** and change it to read **data**.

The previous two corrections were done in overwrite mode: the typed characters replaced the previous characters. The next correction will be done in insert mode.

#### Inserting

To turn on insert mode, press **Ctrl** + **I**. This will light up the insert indicator on the bottom of the screen. With this indicator lit, every character typed will be inserted at the cursor and all following characters will be pushed to the right.

The third and last correction is adding the word **UniSite** before the period below the word **loaded**. Move the cursor on top of the period and type

**UniSite**

## Restoring Your File

If you are not satisfied with your edits and you want to restore the current screen of data, press **Ctrl** + **U** . UniSite restores the current screen of data to the state it was in after the last save.

---

*Note: When you move to another block, all edits are saved.*

## Saving Your Edits

Save your edits if you are satisfied with them. Press **F2** to save your edits and return to the previous menu.

---

*Note: When you exit the Edit screen, or move to another block, all edits are saved. If saving to RAM, there is no way to recall the original data file once you save an edited data file.*

## Review

In this Session, you learned how to edit a data file stored in RAM. To access UniSite's built-in editor, press **M** **E** **E** from the Main Menu. Then, from the Edit Programmer Memory screen, select **R** to edit RAM. Once you are in the editor, you can edit data in either its hex representation or its ASCII representation. Press **Tab** to toggle between the two modes.

The editor defaults to overwrite mode, but it can also operate in insert mode. Press **Ctrl** + **T** to toggle between the two modes.

After making edits to a page of data, you can restore the page to its original condition by pressing **Ctrl** + **U** .

## Session 10: Programming a Memory Device

This Session shows you how to use UniSite to program a memory device.

### Before You Begin

Make sure you have completed Session 1, an introduction to the UniSite interface.

*Note: During this Session, you will program a device with a sample data file stored on the Algorithm disk. Normally, you would want to program a device with data stored in RAM because RAM operations are much faster than disk operations.*

### Can I Use Another Device?

If you do not have an AMD 27256 (the device we are going to use for this Session), then go to the Select Device screen and select the device you are going to program. Keep in mind that the device you select might not have the same capabilities as the AMD 27256. For example, the AMD 27256 supports Electronic ID while the Hitachi 27256 does not.

### Parameter Screens

Select Program Device from the Main Menu. The dialog window displays the Program Memory Device screen, shown in Figure 4-19.

**Figure 4-19**  
The Program Memory  
Device Screen

FILENAME: _____		RAM AVAIL: 128 OF 128KB		REV: X.XX Y.YY Z.Z	
MANUFACTURER: AMD		PART #: 27256		FAMILY/PIN CODE: 0C1 / 03Z	
I/O FORMAT: Intel Intellec 8/MDS					
MAIN MENU		PROGRAM MEMORY DEVICE (non-default)			
Select device	Source (RAM, Disk)	N			
Quick copy	Data word width	8			
Load device	Next device	1			
<b>Program device</b>	Total set size	1			
Verify device	User data size	0000			
More commands	Next operation begins at	0			
		Return: Execute F4: Select mode/options			
		F1: Main menu F2: Prev menu F3 or ? : Help			

Look at the Program Memory Device screen. At the top of the dialog window, you see one of the following:

PROGRAM MEMORY DEVICE (all parameters)

PROGRAM MEMORY DEVICE (non-default)

There are two types of programming parameter screens: simple and complex. UniSite defaults to displaying the simple, Non-default parameters screen.



## What Is the Difference?

Both screens let you change programming parameters. The All Parameters screen contains all the programming parameters supported by the selected device. While the Non-default parameters screen contains a simplified subset of the programming parameters supported by the selected device.

To see the difference between the two screens, press **[F4]**. If you were looking at the Non-default screen, you are now looking at the All Parameters screen. (Likewise, if you were looking at the All Parameters screen, you are now looking at the Non-default screen.) As you can see, there is roughly a 2:1 difference in the number of parameters between the two screens. The All Parameters screen is shown in Figure 4-20.

**Figure 4-20**  
The All Parameters Screen

FILENAME:		RAM AVAIL: 128 OF 128KB		REV: X.XX Y.YY Z.Z	
MANUFACTURER: AMD		PART #: 27256		FAMILY/PIN CODE: 0C1 / 032	
I/O FORMAT: Intel Intellec 8/MDS					
MAIN MENU		PROGRAM MEMORY DEVICE (all parameters)			
Select device	Source (RAM.Disk)	<input type="checkbox"/>	Set auto-increment		
Quick copy	Data word width	<input type="checkbox"/>	Illegal bit chk		
Load device	Next device	<input type="checkbox"/>	Blank check		
Program device	Total set size	<input type="checkbox"/>	Compare elec ID		
Verify device	User data size	<input type="checkbox"/>	Enable yield tally		
More commands	Next operation begins at	<input type="checkbox"/>	Reject option (C,S)		
	Memory begin address	<input type="checkbox"/>	Verify passes (0,1,2)		
	Device begin address	<input type="checkbox"/>			
	Device block size	<input type="checkbox"/>			
Return: Execute		F4: Select mode/options			
F1: Main menu		F2: Prev menu F3 or ?: Help			

## About Parameters

Parameters are user-definable fields that determine what UniSite does. Parameters either qualify or quantify UniSite's actions. Qualifying parameters, such as Illegal Bit Check, control whether or not UniSite performs an operation. Quantifying parameters, such as Block Size, give UniSite a range or variable to use in an operation.

## Which Screen Should I Use?

During the course of normal programming, you usually only need the parameters on the Non-default screen. But if you want to do some more complicated programming operations, you need to change some of the parameters on the All Parameters screen. A complicated programming operation might be if you wanted to program only part of a device.

For this Session, you should be looking at the Non-default screen. If the All Parameters screen is displayed, press **[F4]** to switch to the Non-default screen.

## Set the Programming Parameters

Look at the Non-default parameters screen and set the programming parameters as follows:

- Source - D (disk)
- Filename - sample.bin

When you enter the filename, UniSite searches the disk in the drive for the source data file. When it finds the file, UniSite sets the User Data Size and Memory block parameters according to the data in the data file. When UniSite has set the parameters, it displays the following message in the message bar:

Memory block parameters now set for data file operation

The Program Memory Device screen should now look like the screen shown in Figure 4-21.

**Figure 4-21**  
*The Programming Parameters for This Session*

```

FILENAME: SAMPLE.BIN      RAM AVAIL: 128 OF 128KB  REV: X.XX  Y.YY  Z.Z
MANUFACTURER: AMD        PART #: 27256          FAMILY/PIN CODE: 8C1 / 032
I/O FORMAT: Intel Intellec 8/MDS
- Memory block parameters now set for data file operation.

MAIN MENU                PROGRAM MEMORY DEVICE  (non-default)
Select device            Source (RAM,Disk)
Quick copy               Filename                SAMPLE.BIN
Load device              Data word width        8
                          Next device            1
Program device           Total set size          1
Verify device            User data size         0
More commands            Next operation begins at 0

                          Return: Execute      F4: Select mode/options
                          F1: Main menu        F2: Prev menu      F3 or ? : Help
  
```

## Programming the Device

Now that you have set the programming parameters, you are ready to program the data file into the device. To begin programming, press **[J]**. After UniSite has programmed the device, the following message appears in the message bar:

OPERATION COMPLETE: Sumcheck = xxxxxxxx

where xxxxxxxx represents is the 8-digit sumcheck of the data programmed into the device in the socket.

## Review

When you select Program Device from the Main Menu, you see one of two parameter-entry screens: the simple Non-default screen or the complex All Parameters screen. Press **[F4]** to toggle between the two parameter-entry screens.

When you have entered the programming parameters, press **[J]** to begin the programming. When UniSite is finished programming, it displays an 8-digit sumcheck of the data programmed into the device. (UniSite displays a 4-digit sumcheck when you program a logic device.)

## Session 11: Verifying a Device

Once you have programmed a device, you can perform a number of different device checks and programming tests on the device. For the rest of this Session we will refer to device checks and programming tests as verify operations.

*Note: During this Session, you will verify a device with a data file stored on the Algorithm disk. Normally, you would want to verify a device with data stored in RAM because RAM operations are much faster than disk operations.*

### Before You Begin

Make sure you have completed Session 1, an introduction to the UniSite interface. Also make sure you have completed Session 9, which covers programming a device.

### Can I Use Another Device?

If you do not have an AMD 27256 (the device we are going to use for this Session), then go to the Select Device screen and select the device you are going to program. Keep in mind that the device you select might not have the same capabilities as the AMD 27256. For example, the AMD 27256 supports Electronic ID while the Hitachi 27256 does not.

### Parameter Screens

Select Verify Device from the Main Menu. The dialog window displays the Verify Memory Device screen, shown in Figure 4-22.

**Figure 4-22**  
The All Parameters Verify Screen

FILENAME:                      RAM AVAIL: 128 DF 128KB                      REV: X.XX   Y.YY   Z.Z MANUFACTURER: AMD                      PART #: 27256                      FAMILY/PIN CODE: 0C1 / 03Z I/D FORMAT: Intel Intellec 8/MDS	
<b>MAIN MENU</b> Select device Quick copy Load device Program device <b>Verify device</b> More commands	<b>VERIFY MEMORY DEVICE</b> (all parameters) Source (RAM, Disk) <b>R</b> Data word width <b>8</b> Next device <b>1</b> Total set size <b>1</b> User data size <b>8000</b> Next operation begins at <b>0</b> Memory begin address <b>0</b> Device begin address <b>0</b> Device block size <b>8000</b> Set auto-increment <b>N</b> Compare elec ID Verify passes (1,2) <b>2</b> Return: Execute                      F4: Select mode/options F1: Main menu                      F2: Prev menu                      F3 or ? : Help

Look at the Verify Memory Device screen. At the top of the dialog window, you see one of the following:

VERIFY MEMORY DEVICE (all parameters)

VERIFY MEMORY DEVICE (non-default)

As with the programming parameters screens, the parameter screens let you change programming parameters. The All Parameters screen contains all the verify parameters supported by the selected device. While the Non-default parameters screen contains a simplified subset of the verify parameters supported by the selected device.

## Which Should I Use?

During the course of verifying a device, you will usually need only the parameters on the Non-default screen. But if you want to do some more complicated verify operations, you will need to change some of the parameters on the All Parameters screen. A complicated verify operation might involve, for instance, verifying only part of a device.

For this Session, you should be looking at the Non-default screen. If the All Parameters screen is displayed, press **[F4]** to switch to the Non-default screen, which is shown in Figure 4-23.

**Figure 4-23**  
The Non-default Verify Parameters Screen

```

FILENAME: SAMPLE.BIN      RAM AVAIL: 128 OF 128KB  REV: X.XX  Y.YY  Z.Z
MANUFACTURER: AMD        PART #: 27256          FAMILY/PIN CODE: 0C1 / 032
I/O FORMAT: Intel Intellec 8/MDS
- Memory block parameters now set for data file operation.

MAIN MENU      VERIFY MEMORY DEVICE      (non-default)

Select device   Source (RAM, Disk)      0
                Filename                SAMPLE.BIN

Quick copy      Data word width        8
                Next device            1

Load device

Program device

Verify device   Total set size          1
                User data size        8000
                Next operation begins at 0

More commands

                Return: Execute      F4: Select mode/options
                F1: Main menu        F2: Prev menu      F3 or ? : Help
  
```

## What Happens When I Change Parameters?

Look at the User data size field. If you just completed Session 10: Program a Memory Device, then this field should be set to C0. Normally, User data size defaults to the size of the selected device. But if you have performed a previous device operation, UniSite sets this value, and other values, to the parameters specified in the previous device operation. Other parameters affected are Block Size and Begin Address.

## Setting the Verify Parameters

If you just completed the previous Session, Program a Memory Device, and then continued immediately with this Session, you probably do not need to set any parameters. But, to be sure, check the parameters shown in Figure 4-23 against the parameters you see on your UniSite.

*Note: Remember that the displays may look different if you are using a device other than an AMD 27256.*

## Verifying the Device

Now that you have set the verify parameters, you are ready to verify the data programmed into the device.

To begin the verify, press **[J]**. If the verify operation completes successfully, the following message is displayed in the message bar:

```
OPERATION COMPLETE: Sumcheck = xxxxxxxx
```

where xxxxxxxx represents is the 8-digit sumcheck of the data in the device.

## Review

When you select Verify Device from the Main Menu, you see one of two parameter-entry screens: the simple Non-default screen or the complex All Parameters screen. Press **F4** to toggle between the two parameter-entry screens.

When you have entered the verify parameters, press **↵** to begin the verify.

When UniSite is finished verifying, it displays an 8-digit sumcheck of the data in the device. (UniSite displays a 4-digit sumcheck when you verify a logic device.)

# 5 Commands

---

This chapter describes the commands you can access from UniSite's menus.

---

## Menu Organization

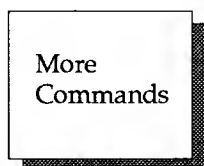
The interrelation of UniSite menus and commands is shown in the command tree in Figure 5-1.

### Menu Maps

Each command description includes a map, part of the command tree representing your location and showing you the path to the command.

The maps, read from left to right, are comprised of three or four boxes, each representing a screen type, as described below.

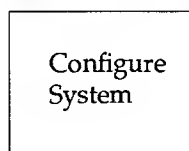
#### Shaded Box



#### Main Menu Screen

A shaded box represents a top-level command, either a Main Menu command or the More Commands menu. These are the most frequently used commands. The shaded box is the start of your path to a particular command.

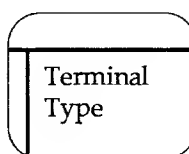
#### Box



#### Other Menu Screens

The next step on the path to a command is a box, representing all other menu screens, most of which are accessible from the More Commands screen. When you select an item in a box, a T-box item or another screen is displayed.

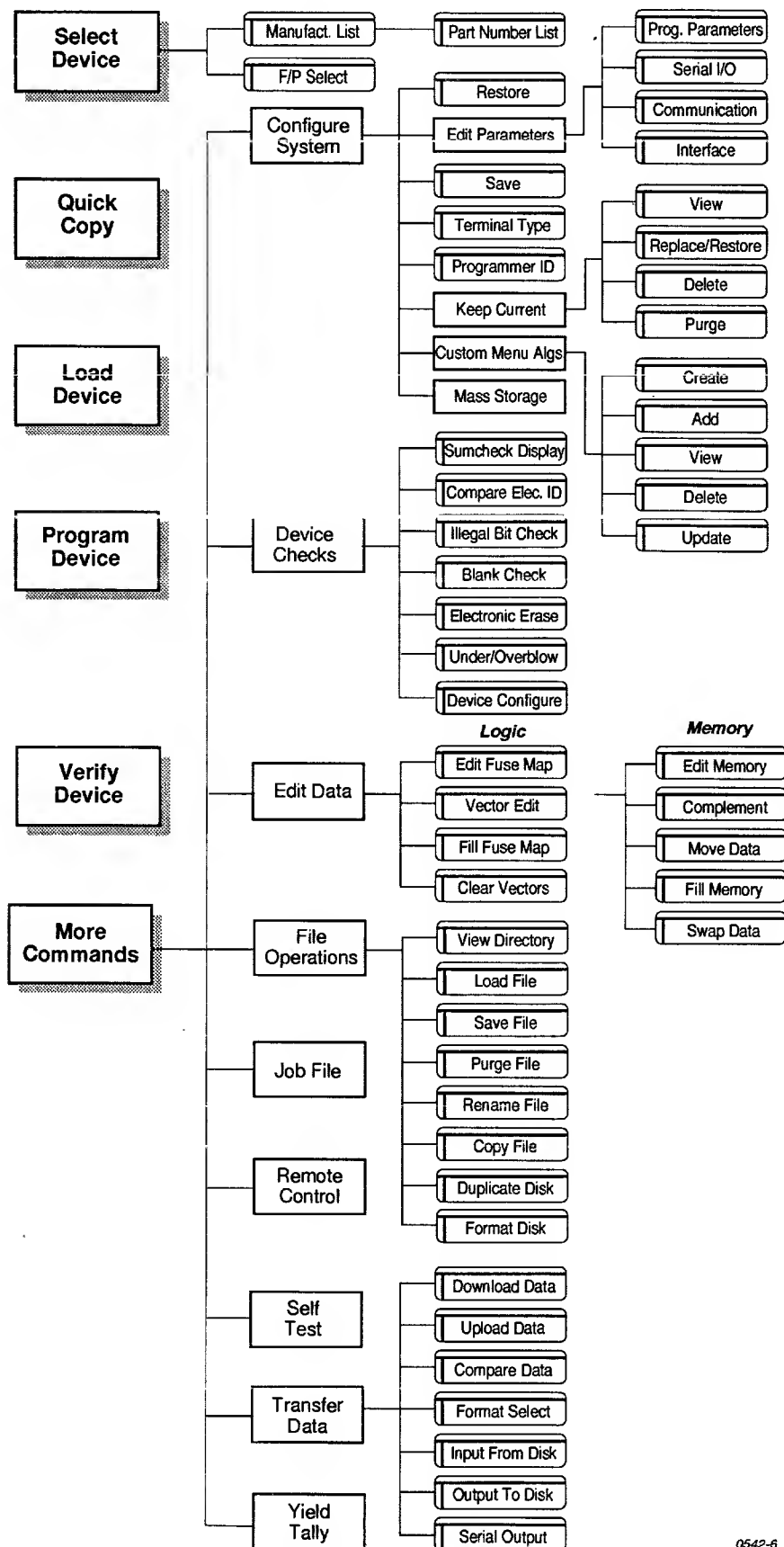
#### T-Box



#### Dialog Window Screen

Selecting the item shown in a T-box brings you to your final destination, a dialog window.

**Figure 5-1**  
The Command Tree



0542-6

## Overwriting User RAM

The following operations use User RAM as a temporary storage buffer, overwriting any data that may have been there previously.

- Load device to disk file
- Program device from disk file
- Verify device from disk file
- Duplicate disk
- Copy file
- Serial output from disk file
- Download data to disk file
- Self-Test User RAM
- Sumcheck Display from disk file
- Illegal Bit Check to disk file
- Underblow to disk file
- Upload data from disk file
- Output data to disk from disk file
- Input data from disk to disk file
- Compare data to disk file
- Create and Add Custom Menu Algorithms

## Factory Default Settings

UniSite's system parameters are initialized to certain settings at the factory. You can restore these factory defaults at any time by selecting configuration file number 0 from the More Commands/Configure System/Restore menu. From the Restore System Parameters menu, press **0** **↓** to re-elect the factory settings. The power-up defaults (configuration file number 1) are the same as the factory defaults when the unit is shipped from Data I/O.

Parameter	Factory Default Setting
Algorithm Type	D
Blank Check	Yes
Continuity Check	Yes
Compare Electronic ID	Yes
Data source/destination	RAM
Data Word Width	8
Device Begin Address	0
Device Block Size	1000
Display device footnote	Yes
Enable download echo	No
Enable security fuse	No
Enable special data	No
Enable terminal beep	Yes
Enable yield tally option	No
EE bulk erase option	No
EOF delimiter flag (download)	No
EOF delimiter flag (upload)	No
Family code	(no default)
File delimiter character (download)	1A (Ctrl-Z)



Parameter	Factory Default Setting
File delimiter character (upload)	1A (Ctrl-Z)
Filename	Blank
Fill RAM before downloading	No
Fill RAM with data (00 to FF)	00
High speed download	Yes
Host command (download)	Blank
Host command (upload)	Blank
Illegal Bit Check	Yes
Instrument control code (0,1,2)	0
I/O Address Offset	FFFFFFFF
I/O translation format	0 (no default)
I/O timeout	30 seconds
JEDEC I/O translate DIP/LCC option	Yes
Logic verification (all, fuse, vector)	All
Main menu job files	No
Manufacturer	Blank (no default)
Memory Begin Address	0
Number of lines between form feeds	0
Number of nulls	0
Odd/Even Byte Swap	No
Part number	(no default)
Pinout code	(no default)
Power on CRC mode	No
Power up user RAM test	Yes
Program security fuse	No
RAM Device Selection	No
Reject option (commercial or single)	Commercial
Remote Off code	0
Remote On code	0
Remote serial port configuration	9600 baud, 8 data bits, no parity, 1 stop bit, active CTS/DTR
Security Fuse Data (0 or 1)	0
Serial set auto increment flag	No
Simple/complex parameter screen	Simple
Terminal serial port configuration	9600 baud, 8 data bits, no parity, 1 stop bit, active CTS/DTR
Terminal type	VT-100
Transmit pacing	0
User Menu Port *	T
User Data Size	0
Upload wait	0 seconds
Upload destination/download source	Remote
Upload record size	16
Verify Data Format	Hex
Verify Passes	2

\* This parameter is not restored when the Restore Configuration operation is performed. However, it is used at power-up time if it is saved as a power-up parameter.

---

## Select Device

Before you can perform any device-related operations with the programmer, you must select the device you are using.

---

*Note: If you are using a new version of software (i.e., you received an update), you must perform the Update operation prior to selecting a device. See page 2-28 for more information.*

*If you select a device before properly updating a new version of software, UniSite displays the following message*

OPERATION ABORTED: Product security violation

### Before You Select a Device

Before selecting a device, set the **Algorithm Type** parameter to the type of algorithm that matches the device you wish to select.

To set the Algorithm Type, from the Main Menu select **More commands/ Configure system/Edit/Programming** (refer to "Algorithm Type" on page 5-31 for more information).

---

*Note: You can press **F1** to return to the Main Menu from any submenu.*

The Programming Parameters screen appears. On this screen, set the Algorithm Type to one of the following:

- **D (default)** — Set the algorithm type to **D** if you wish to select a device from the **Manufacturer List**. The Manufacturer List contains almost all of the devices supported by your programmer (the programming algorithms for these devices reside on your Algorithm disks). Selecting a device from this list is a two-step process: first, select the device manufacturer and then the device part number.
- **E (extended algorithm)** — Set the algorithm type to **E** if you wish to select a device from an **Extended Algorithm** list. An Extended Algorithm list contains devices that use extended algorithms for programming (the programming algorithms for these devices reside on a disk that contains the extended algorithms). Selecting a device from this list is a two-step process: first, select the device manufacturer and then the device part number.

If your system includes an Algorithm/System disk labeled **Archive**, set the algorithm type to **E** to access the algorithms on this disk.

---

*Note: The Archive disk contains algorithms for devices that are older or less frequently used. Periodically, as device support grows, algorithms for devices may be removed from the default (D) Algorithm/System disk(s) to make room for newer devices and technology. If you wish to use an algorithm that no longer appears on the default Algorithm/System disk(s), check to see if it is included on the Archive disk.*

- **K (Keep Current)** — Set the algorithm type to **K** if you wish to select a device from a **Keep Current** list. A **Keep Current** list contains devices that use **Keep Current** algorithms for programming (see "Keep Current" on page 5-46 for more information ).

---

*Note:* You can also access **Keep Current** devices by setting the algorithm type to **D** (default) and selecting **KEEP CURRENT** from the manufacturers list.

- **C (Custom Menu)** — Set the algorithm type to **C** if you wish to select a device from a **Custom Menu** list. A **Custom Menu** list contains devices that you included in a **Custom Menu** (see "Custom Menu Algs" on page 5-50 for more information).

After setting the **Algorithm Type** parameter, press **[F1]** to return to the **Main Menu**.

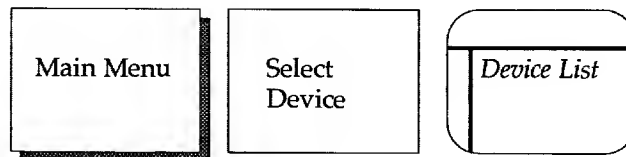
---

*Note:* See "Save System Parameters" on page 5-41 for instructions on how to save your **Algorithm Type** and other parameter changes as power-up defaults.

## Selecting a Device

To select a device, do the following:

1. From the **Main Menu**, select **Select Device**. (If you are not in the **Main Menu**, press **[F1]** to return to the **Main Menu**.)



2. You might be prompted to insert a disk. If so, insert the disk you are prompted for.
3. Depending on which **Algorithm Type** is selected, one of the following screens appears:
  - The **Device Manufacturer List** screen appears if the algorithm type is set to **D** (default). Selecting a device from this screen is a two-step process: first, select the device manufacturer and then the device part number. Continue with steps 4 and 5 of this procedure.
  - The **Extended Algorithm** screen appears if the algorithm type is set to **E** (extended). Selecting a device from this screen is a two-step process: first, select the device manufacturer and then the device part number. Continue with steps 4 and 5 of this procedure.

- The **Keep Current** screen appears if the algorithm type is set to **K** (Keep Current). Skip to step 5 of this procedure.
- The **Custom Menu** screen appears if the algorithm type is set to **C** (Custom Menu). Skip to step 5 of this procedure.

---

*Note: If a different screen than the one you want appears, refer to "Before You Select a Device" on page 5-5 for instructions on how to select an Algorithm Type.*

4. If your Algorithm Type is set to **D** (default) or **E** (Extended Algorithm), a list of device manufacturers supported by the programmer appears. The list also contains the following non-manufacturer choices:

**FAM/PINOUT** Selecting FAM/PINOUT allows you to select devices by entering the family/pinout code directly. However, we recommend that you do not use family/pinout codes to select devices. Use the manufacturer name and part name to select devices.

The family/pinout code feature is supported only for backwards compatibility with older software drivers.

See "Family/Pinout Code Selection" later in this chapter for more information.

**KEEP CURRENT** Selecting KEEP CURRENT allows you to select a device whose algorithm was downloaded using Keep Current. This choice is the same as selecting **K** (Keep Current) as the Algorithm Type.

---

*Note: Depending on what version of software you have, additional non-manufacturer choices may be available.*

Select a manufacturer (or one of the non-manufacturer choices), by locating the manufacturer choice and entering the number appearing next to that choice in the manufacturer field. For example, if 2) KEEP CURRENT is a choice, type the number 2 for the manufacturer and press **Enter**.

When you select a device manufacturer, you can set the device type filter to the appropriate setting for the device you are using. As the names indicate, Logic Only displays only logic devices, Memory & Emicros displays memory and emicros, and All displays all device types. Press **Space** to cycle through the filter types. The list of manufacturers displayed will change based on your choice.

The upper-right corner of the manufacturer screen displays how many screens (pages) of manufacturers there are and what page you are on. Press **Ctrl** + **N** to go to the next page of manufacturers. Press **Ctrl** + **P** to go to the previous page of manufacturers. When the screen repaints, you see the next (or previous) list of manufacturers.

5. A screen appears with a list of devices. To select the device, locate the device you are going to use and enter the number appearing next to that device.

If you do not see the device you are looking for, press **Ctrl** + **N** to advance to the next page of the device list. Press **Ctrl** + **P** if you want to return to the previous page. After you have entered the number for the device you want, press **Enter**. The programmer returns to the Main Menu and the status window shows the device you selected.

If the programmer displays the message `OPERATION ABORTED: Product security violation`, you need to update your programmer to your new version of software. See page 2-28 for more information on updating your new version of software.

## Family/Pinout Code Selection

Family/pinout codes identify devices by their characteristics rather than by their manufacturer and part names.

---

**CAUTION:** *Using family and pinout codes to select a device may lead to damage to your device.*

*Data I/O assumes no responsibility or liability for results produced by entry of family/pinout combinations.*

Due to device-specific changes in programming algorithms by the semiconductor manufacturers and increased device-specific tests performed on the devices, it is no longer recommended that you use family/pinout codes to select devices.

For instance, several similar devices can share the same family/pinout code, but these devices do not always share the same characteristics, because one or more of these devices can undergo process changes that affect their programming characteristics enough to require programmer manufacturers to modify existing algorithms. These changes often take place with little or no identification change to a device because operating characteristics have not changed. Process changes can affect verify and functional tests, as well as direct programming operations. When family/pinout codes are used to select similar devices, a programmer operation done on the device can result in improper programming, corrupted data, or damage to the device.

## How Should I Select Devices?

To maintain high quality, it is recommended that you select devices by manufacturer and part names.

---

**Note:** *The family/pinout code feature might be removed in the future. If you are using family/pinout codes to select devices, it is recommended that you switch to using manufacturer and part names to select devices.*

## After You Select a Device

After you have selected a device, the following occurs:

### Device Information

If there is online device-specific information for the selected device, the programmer automatically displays this information. If there is more than one screen of device specific information, press **Ctrl** + **N** to view the next screen of information. Press **Ctrl** + **P** to view the previous screen of information. Exit the screen by pressing **F1** or **F2**.

If the Display Device Footnotes parameter (in the More commands/Configure system/Edit/Programming screen) is set to N, device-specific information is not displayed, instead the following message appears in the message bar if there is online device-specific information for the selected device.

Hit F3 or ? to view device specific message.

To view the online information, press **F3** or **?**. The screen clears and the programmer displays the device-specific information.

---

*Note: If only the footnote number appears in the message bar, the programmer could not access the **devfnote.sys** file. Check to see if the correct disk is inserted into the programmer's drive.*

*Footnote (device-specific) information can also be found in the **Device List**.*

### Device-related Selections Available

The manufacturer and part number of the device you selected appears in the status window and the programmer displays only screens related to the type of device you selected. For example, if you choose a logic device, you will see only screens that are required to load, program, edit, and verify a logic device.

Also, some devices support special functions, such as the Program Signature and XNOR table. If the device you select supports a particular function, parameter entry fields associated with that function are displayed.

---

## Cross Programming

Cross programming allows a single generic programmable logic device (PLD) to be configured as any one of many PLD architectures.

Consequently, the generic device can take on the function of many subset devices. The term generic PLD is used to identify the superset device, such as a 16V8 generic PLD, which can be configured as a 16R4, 16R8, or a 16L8.

The generic PLD and the subset devices it can support are not restricted to the same manufacturer. For example, a 16V8 generic PLD from manufacturer A can be programmed using a fuse pattern originally designed for a 16L8 from manufacturer B. The cross-programming feature allows you to avoid recompiling source code for the generic PLD if the appropriate fuse pattern is available for a subset part.

To view the subset devices, select a manufacturer with the **XPGM** extension. This screen lists all devices that can be replaced by the generic PLD of the selected manufacturer.

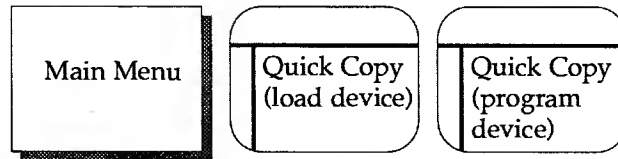
After selecting a device with the **XPGM** extension, load the fuse map of the subset PLD into User Memory, by using either a Load From Device operation or a Download of a JEDEC file.

Select a manufacturer with an **XPGM** extension and press . Then, from the Part Menu for Manufacturer screen, select the appropriate device, such as **16V8** as **16L8** if a 16L8 fuse map has been loaded and a 16V8 generic PLD is to be programmed. After making your selection, press  and the Main Menu appears. Select the Program Devices screen and 16V8 as 16L8 will be displayed in the PART # field at the top of the screen. Insert the 16V8 and press . The 16V8 is programmed as a 16L8.



## Quick Copy

The Quick Copy command allows you to load data from a master device and program that device data into a target device quickly and easily.



---

*Note:* Before you use the Quick Copy command, you must first select a device.

To use the Quick Copy command, follow these steps:

1. Select the device that contains the data you are going to load. For more information, see the explanation of the Select Device command earlier in this chapter.
2. Select the Quick Copy command by pressing **[Q]** from the Main Menu. The dialog window displays the Quick Copy screen. The message bar displays the following message:

Insert master device. Hit return.

3. Insert the master device and lock it into place.
4. Press **[J]** to begin loading the master data into RAM. When the data in the master device has been loaded, UniSite displays

OPERATION COMPLETE. Sumcheck = xxxxxxxx Hit return

---

*Note:* If you are using the Quick Copy function with SetSite, you must press Return again at this point. This allows you time to review the screen and sumcheck information.

5. Remove the master device and press **[J]**. UniSite displays
- Insert blank device. Hit return.
6. Insert the target device (the device you want to program) and lock it into place.
  7. Press **[J]** to program the device.
  8. When the programming operation is complete, UniSite displays
- OPERATION COMPLETE. Sumcheck = xxxxxxxx Hit return
9. The Quick Copy operation is complete. Remove the device.
  10. To program another device, return to step 6.

---

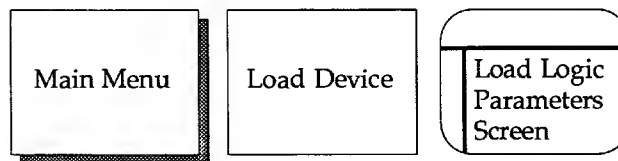
*Note:* You may also use the Quick Copy to load data for gang programming, if you are using the SetSite module. Data Word Width will automatically be set to Device Word Width.

## Load Device

The Load Device command allows you to copy data from a master device into either a disk file or into RAM. Depending on whether you select a logic or memory device, either the Load Logic Device screen or the Load Memory Device screen will appear when you select this command. The Load Logic Device command is explained first, and the Load Memory Device command is explained after that.

### Load Logic Device

If you select a logic device and then select the Load Device command, the Load Logic Device menu will appear.



To load user memory with data from a logic device, follow these steps:

1. Select and socket a logic device.
2. Select Load Device from the Main Menu. The Load Logic Device screen appears.
3. Specify the parameters you want and press **[J]** to begin the loading.
4. When the load operation is complete, UniSite displays the following message in the message bar:

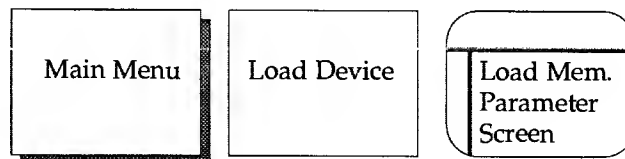
OPERATION COMPLETE. Sumcheck = xxxx

The available parameters are described below.

- **Destination (R,D)** — Specifies the destination for the fuse data you will be loading. Press **[Space]** to toggle between **R** (RAM) and **D** (Disk).
- **Filename** — Specifies the name of the disk file where you want the data to be saved. This option appears only if you specify disk as the Destination. The filename must follow standard DOS conventions, and may contain a drive designator. An example of a valid filename is **b:16r8.dat**.

## Load Memory Device

If you select a memory device and then select the Load Device command, the Load Memory Device menu will appear.



To load user memory with data from a memory device, follow these steps:

1. Select and socket a memory device.
2. Select Load Device from the Main Menu. The Load Memory Device screen appears.
3. Specify the parameters you want. Then, press  to begin the loading.
4. When the load operation is complete, UniSite displays the following message in the message bar:

OPERATION COMPLETE. Sumcheck = xxxxxxxx

The following parameters can be specified on the Load Memory Device screen:

- **Destination (R,D)** — Specifies the destination for the data you will be loading. Press  to toggle between R (RAM) and D (Disk).
- **Filename** — Specifies the name of the disk file where you want the fuse data to be saved. This option appears only if you specify disk as the Destination. The filename must follow standard DOS conventions, and may contain a drive designator. An example of a valid filename is b:27256.dat.
- **Data Word Width** — Sets the word width of data to be loaded. For 8-bit (or larger) devices, the minimum word width is equal to the device width, and the maximum word width is 64. For 4-bit devices, your word width choices are 4, 8, 16, and 32. This value should match the data bus word width in the target system for the device being programmed. The default value is the device word width.
- **Next Device** — Designates the next device (next set member) in the set. For example, if you are using 8-bit devices and have specified a word width of 16 bits on the Load Memory Device screen, it requires two devices to store each 16-bit word. Typing  for the next set member directs UniSite to load the first device in the set at even-address bytes of the memory block. Typing  directs UniSite to load the second device at odd address bytes of the memory block.

- **Total Set Size** — Specifies how many virtual devices are in the set for device operations. For example, if you are loading 16-bit wide data from two 8-bit wide devices, your virtual device still equals one (one 16-bit virtual device). Any number between 1 and 99 can be entered. Automatic Set Size calculation will be attempted when any of the following parameters are changed: Device width, Data Word Width, Device Block Size or User Data Size. Total Set Size can be defined by the following equation:

$$\text{TOTAL SET SIZE} = \frac{\text{USER DATA SIZE}}{\text{DEVICE BLOCK SIZE} \left( \frac{\text{DATA WORD WIDTH}}{\text{DEVICE WIDTH}} \right)}$$

0987-1

- **User Data Size** — Specifies the hexadecimal size, in bytes, of the data block used to load from the device to the destination. This value is normally equal to the device size or to a multiple of the device size for loading a set. If 0 or a value less than the device size is entered, it is reset to the device size for the load. User Data Size works with Total Set Size to determine the total amount of bytes to load from a set of devices.
- **Next Operation Begins At** — This read-only parameter shows where in user memory the next data byte will be loaded. This value is calculated from the Data Word Width, Device Block Size, Memory Begin Address, device width, and next set member parameters.

#### Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite defaults to displaying the simple, Non-default Parameters screen. If you want to view the complex, All Parameters screen, press **[F4]** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also show up on the Non-default Parameters screen.

The following parameters are found on the complex, all parameters Program Memory Device screen:

- **Memory Begin Address** — Specifies the first address, in hex, where the first byte of data is loaded from a device. If the destination is RAM, it is a beginning RAM address. If the destination is Disk, it is a beginning disk file address. The Memory Begin Address must be an even address if you have selected a 16-bit device. The default address is 0.
- **Device Begin Address** — Specifies the first hex master device address that will be loaded. The default value is 0.
- **Device Block Size** — Specifies the size, in hex, of device data used in device operations. After selecting a device, the Device Block Size is set automatically to the device size and normally does not need to be changed. Also, Device Block Size is set to a smaller value if the Device Begin Address is nonzero. This parameter can be changed if desired. Entering a zero sets the Device Block Size equal to the device size.

- **Set Auto-increment** — This option, used in serial set mode, automatically directs UniSite to the next block in the set that is to be loaded. For example, if you have four 1K x 8 devices to load into a 4K x 8 block of memory, using the auto-increment option directs UniSite to point to the first memory address of the next 1K block after each device had been loaded. For single device operations, this feature should be disabled and the Next Device parameter should be set to 1.

---

*Note: Items with an \* (asterisk) are visible on the screen only if the selected device supports the feature.*

- \* **Compare Electronic ID (Y,N)** — Compares the electronic ID of the socketed device against the electronic ID of the selected algorithm.
- \* **Odd/Even Byte Swap (Y,N)** — When enabled, allows the Most Significant Bytes (MSB) and the least significant bytes (LSB) of 16-bit words to be swapped when data is loaded from a 16-bit device. When disabled, the data from a 16-bit device is loaded into User RAM with the MSB stored at an odd memory address. When enabled, the MSB is stored at an even memory address.

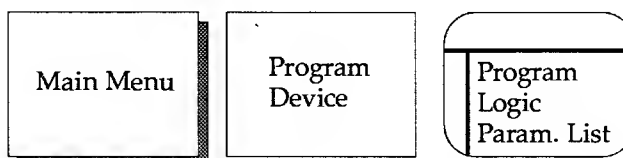
## Program Device

The Program Device command allows you to copy data from RAM or from disk into a blank device. Depending on whether you select a logic or memory device, either the Program Logic Device screen or the Program Memory Device screen appears when you select this command. The Program Logic Device command is explained first, and the Program Memory Device command is explained next.

Before you can program a device, you need to load the programming data into RAM or select a disk file as your data source.

### Program Logic Device

If you select a logic device and then select the Program command, the Program Logic Device menu appears.



To program a logic device, follow these steps:

1. Select and socket a logic device.
2. Select Program Device from the Main Menu. The Program Logic Device screen appears.
3. Specify the parameters you want. Then press **[J]** to begin the programming.
4. When the programming is complete, UniSite displays the following message in the message bar:

OPERATION COMPLETE. Sumcheck = xxxx

The following parameters can be specified on the Program Logic Device screen.

- **Source (R,D)** — Specifies the source of the fuse data and test vectors that you wish to use. Press **[Space]** to toggle between R (RAM) and D (Disk).
- **Filename** — Specifies the name of the file you want as your data source. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and may contain a drive designator. An example of a valid filename is **b:16r8.dat**.
- **Security Fuse Data (0,1)** — To program the security fuse, set this parameter to 1 and set the Program Security Fuse parameter to Y. This parameter defaults to 0, which disables programming of the security fuse.

- **Program Security Fuse (Y,N)** — Enables or disables the programming of the security fuse. To program the security fuse, set this parameter to Y and set the Security Fuse Data parameter to 1. This parameter defaults to N, which disables programming of the security fuse.

### Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite defaults to displaying the simple, Non-default Parameters screen. If you want to view the complex, All Parameters screen, press **[F4]** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also appear on the Non-default Parameters screen.

The following parameters are found on the complex, all parameters Program Logic Device screen.

- **Illegal Bit Check (Y,N)** — Enables or disables the illegal-bit test. When enabled, this test compares data in a device against data in UniSite's RAM to determine if the device has already-programmed locations of incorrect polarity. For example, UniSite returns an illegal-bit error in the following situation: data in RAM indicates a specific bit should be in an unprogrammed state while the corresponding bit in the device is in a programmed state. The device cannot be programmed if UniSite detects an illegal bit. This parameter is enabled by default.
- **Blank Check (Y,N)** — Enables or disables the blank check test. A Blank Check test checks a device for programmed bits. This parameter defaults to Y, which enables the test.
- **Enable Yield Tally (Y,N)** — When enabled, directs UniSite to keep a running tally of the programming yields for the last sixteen types of devices programmed. These totals show how many devices passed and failed, and what specific errors, if any, have occurred. This parameter defaults to N, which disables the test. A more complete description of this feature can be found in the "Yield Tally" section of this chapter.
- **Logic Verification (A,F,V)** — Specifies the type of logic verification to perform after programming. Select from Fuse Verification **F**, Vector Verification **V**, or All **A**. Fuse Verification checks the fuse pattern programmed into the device with the pattern in UniSite's memory. Vector Verification functionally tests the device, using structured test vectors stored in memory. **A** directs UniSite to perform both fuse verification and vector verification. Press **[Space]** to step through the three choices. This parameter defaults to **A**.

UniSite does not support vector testing for logic devices with more than 84 pins. Attempting to perform a vector test on a device with more than 84 pins will yield the following message:

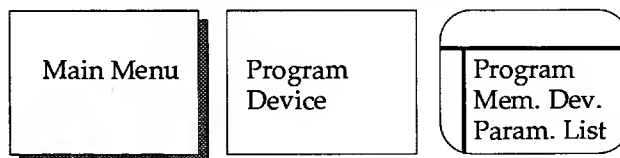
```
OPERATION COMPLETE. Sumcheck = hhhhhhhh (Vector test
not supported)
```

- **Verify Passes (0,1,2)** — Selects the number of times to test the device. **0** directs UniSite not to test the device. **1** directs UniSite to test the device once at the device manufacturer's nominal Vcc. **2** directs UniSite to verify the device at the device manufacturer's recommended high and low Vcc levels. This parameter defaults to **2**.
- **Reject Option (C,S)** — Selects the number of times the device is pulsed with programming voltage before it is rejected as unprogrammable. **C** selects the number of pulses specified by the manufacturer. **S** selects either a one-pulse or the military-specification number of programming pulses. Unless you are programming devices to a strict military specification, you should leave this option set at **C**. This option defaults to **C**.

## Program Memory Device

If you select a memory device and then select the Program command, the Program Memory Device screen appears.

*Note: If device configuration is supported for the selected device, you can set the program, erase, and protect separately for each sector using the More Commands/Device Checks/Device Configure command.*



To program a memory device, follow these steps:

1. Select and socket a memory device.
2. Select Program Device from the Main Menu. The Program Memory Device screen appears.
3. Specify the parameters you want. Press  to begin the programming.
4. When the programming is complete, UniSite displays the following message in the message bar:

OPERATION COMPLETE. Sumcheck = xxxxxxxx

The following parameters can be specified on this screen:

- **Source (R,D)** — Specifies the source for the data to program into the device. Press  to toggle between R (RAM) and D (disk).
- **Filename** — Specifies the name of the file you want as your data source. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and may contain a drive designator. An example of a valid filename is **b:27512.dat**.
- **Data Word Width** — Sets the word width of the data to be programmed. For 8-bit (or larger) devices, the minimum word width is equal to the device width and the maximum is 64. For 4-bit devices, your word width choices are 4, 8, 16, and 32. This value should match the data bus word width in the target system for the device being programmed.



- **Next Device** — Designates the next device in the set. For example, if you are using 8-bit devices and have specified a word width of 16 bits on the Program memory device screen, it requires two devices to store each 16-bit word. **1** directs UniSite to program the first device in the set with even-numbered addresses of the memory block. **2** directs UniSite to use odd-numbered addresses.
- **Total Set Size** — Specifies how many virtual devices are in the set for device operations. For example, if you are programming 16-bit wide data into two 8-bit wide devices, your virtual device still equals one (one 16-bit virtual device). You can enter any number between 1 and 99. Automatic Set Size calculation is attempted when you change any of the following parameters: Device width, Device Block Size or User Data Size. Total Set Size is defined by the following equation:

$$\text{TOTAL SET SIZE} = \frac{\text{USER DATA SIZE}}{\text{DEVICE BLOCK SIZE} \left( \frac{\text{DATA WORD WIDTH}}{\text{DEVICE WIDTH}} \right)}$$

0987-1

- **User Data Size** — Specifies the hexadecimal size, in bytes, of the data block to program into a device. Normally, this value is equal to the device size or to a multiple of the device size for set programming. Entering 0 sets the User Data Size to the device size. User Data Size works with Total Set Size to determine the total amount of bytes to program into a set of devices.
- **Next Operation Begins At** — This read-only parameter shows what address in user memory contains the next data byte to be programmed. This value is calculated from the Data Word Width, Device Block Size, Memory Begin Address, device width, and next set member parameters.

---

*Note: Items with an \* (asterisk) are visible on the screen only if the selected device supports the feature.*

- \* **Security Fuse Data (0,1)** — To program the security fuse, set this parameter to **1** and set the Program Security Fuse parameter to **Y**. This parameter defaults to **0**, which disables programming of the security fuse.
- \* **Program Security Fuse (Y,N)** — Enables or disables the programming of the security fuse. To program the security fuse, set this parameter to **Y** and set the Security Fuse Data parameter to **1**. This parameter defaults to **N**, which disables programming of the security fuse.
- \* **Program Signature** — Available on only a few devices, the program signature is a user-definable field that allows the user to program data into the program signature array.
- \* **Software Data Protection (Y,N)** — When enabled, prevents writing to a device.

## Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite defaults to displaying the simple, Non-default Parameters screen. If you want to view the complex, All Parameters screen, press **[F4]** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also appear on the Non-default Parameters screen.

The following parameters are found on the complex, all parameters Program Memory Device screen.

- **Memory Begin Address** — Specifies the first address, in hex, of the first byte of data to be programmed. For a SetSite operation, Memory Begin Address specifies the first address of the device in the first socket. If the Source is RAM, it is a beginning RAM address. If the Source is Disk, it is a beginning disk file address. The Memory Begin Address must be an even address if you have selected a 16-bit device. The default address is 0.
- **Device Begin Address** — Specifies the first hex device address that will be programmed. The default is 0.
- **Device Block Size** — Specifies the size, in hex, of device data used in device operations. At device selection, Device Block Size is set to the device size and normally does not need to be changed. It is also automatically set to a smaller value if the Device Begin Address is nonzero. This parameter can be changed if desired. Entering zero sets the Device Block Size equal to the device size.
- **Set Auto-increment** — When enabled, directs UniSite (in serial set programming mode) to the starting memory address of the next block in the set to be programmed. For example, if you have four 1K x 8 devices to program from a 4K x 8 block of data, using the auto-increment option directs UniSite to point to the first address of the next 1K block after each device has been programmed. For single device operations, this feature should be disabled and the Next Device parameter should be set to 1.

---

*Note: Items with an \* (asterisk) are visible on the screen only if the selected device supports the feature.*

- \* **Illegal Bit Check (Y,N)** — Enables or disables the illegal-bit test. This test compares data in a device against data in UniSite's RAM to determine if the device has already-programmed locations of incorrect polarity. For example, UniSite returns an illegal-bit error in the following situation: data in RAM indicates a specific bit should be in an unprogrammed state while the corresponding bit in the device is in a programmed state. The device cannot be programmed if UniSite detects an illegal bit. This parameter is enabled by default.
- **Blank Check (Y,N)** — Enables or disables the blank check test. A Blank Check test checks a device for programmed bits. This parameter defaults to Y, which enables the test.
- \* **Compare Electronic ID (Y,N)** — When enabled, compares the electronic signature of the device against the electronic signature of the selected algorithm.

- **Enable Yield Tally (Y,N)** — When enabled, directs UniSite to keep a running tally of the programming yields for the last sixteen types of devices programmed. These totals show how many devices passed and failed, and what specific errors, if any, occurred. This parameter defaults to N, which disables the test. A more complete description of this feature can be found in the "Yield Tally" section of this chapter.
- \* **Odd/Even Byte Swap (Y,N)** — When enabled, allows the Most Significant Bytes (MSB) and the Least Significant Bytes (LSB) of 16-bit words to be swapped when data is programmed into a 16-bit device. The data is programmed into a device retrieving the Most Significant byte from an odd memory address when the flag is N and an even memory address when it is Y.
- **Reject Option (C,S)** — Selects the number of times the device is pulsed with programming voltage before it is rejected as unprogrammable. C selects the number of pulses specified by the manufacturer. S selects either a one-pulse or the military-specification number of programming pulses. Unless you are programming devices to a strict military specification, you should leave this option set at C. This option defaults to C.
- **Verify Passes (0,1,2)** — Selects the number of times to test the device. 0 directs UniSite not to test the device. 1 directs UniSite to test the device once at the device manufacturer's nominal Vcc. 2 directs UniSite to verify the device at the device manufacturer's recommended high and low Vcc levels. This parameter defaults to 2.
- \* **Erase EE Device (Y,N)** — Allows you to erase electronically-erasable PROMs. Before the programming cycle, UniSite checks the device and displays a warning if the device is non-blank. If you enable the erasing of the device, UniSite erases the device before programming the device.

## Enhanced Security Fuse Capability

The enhanced security fuse capability for EMICRO parts allows Security Fuse Data to be stored in a data file. Currently, some devices support this capability, including the Intel 8742AH. For more information, or to see if a device supports this capability, see the device manufacturer's data book.

### Security Fuse Data Field

The Security Fuse Data field cannot be restored by using the More Commands/Configure System/Restore command. Instead, Security Fuse Data will be restored from a data file.

### General Security Fuse Information

Whenever a disk file is specified as the data source, the programming screen should display the Security Fuse Data from the disk file. When the Security Fuse Data is changed, the data in the disk file is updated. (This applies to both logic and EMICRO devices with security fuses.)

When the security fuse is changed with a disk file as the data source, and a disk error, such as `Disk full`, occurs during the update, the entry is not successful and the old value remains. For memory devices, if the disk file is not big enough to cover the location for security fuses, the file defaults to the unprogrammed state of the fuses.

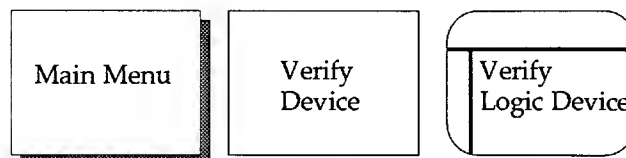
## Verify Device

The Verify Device command compares data in a programmed device with data in RAM or in a disk file. Depending on which type of device is selected, either the Verify Logic Device screen or the Verify Memory Device screen appears when you select the Verify Device command.

Before you can verify a device, you need to load the data into RAM or select a disk file as your data source. RAM data can be loaded from a master device, from a disk file, or through the serial port.

### Verify Logic Device

If you select a logic device and then select the Verify command, UniSite's Verify Logic Device screen appears.



To verify a logic device, follow these steps:

1. Select and socket a logic device.
2. Specify the parameters you want. Then press  to begin the verifying.
3. When the verify operation is complete, UniSite displays the following message in the message bar:

OPERATION COMPLETE. Sumcheck = xxxx

The following parameters can be specified in the Verify Logic Device screen.

- **Source (R,D)** — Specifies the source of the fuse data and test vectors that you wish to verify. Press  to toggle between **R** (RAM) and **D** (disk).
- **Filename** — Specifies the name of the disk file you want the fuse data and test vectors to be verified with. This option appears only if you specify disk as the Source. The filename parameter must follow standard DOS conventions, and can include a drive descriptor. An example of a valid filename is **b:16r8.dat**.

### Optional Parameter Screens

There are two types of parameter-entry screens: simple and complex. UniSite defaults to displaying the simple, Non-default parameters screen. If you want to view the complex, All Parameters screen, press **[F4]** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also show up on the Non-default Parameters screen.

The following parameters are found on the complex, All Parameters Verify Logic Device screen.

- **Logic Verification (A,F,V)** — Specifies the type of logic verification to perform. Select from Fuse Verification **F**, Vector Verification **V**, or All **A**. Fuse Verification checks the fuse pattern programmed into the device with the pattern in UniSite's memory. Vector Verification functionally tests the device, using structured test vectors stored in memory. **A** directs UniSite to perform both fuse verification and vector verification. Press **[Space]** to step through the three choices. This parameter defaults to **A**.

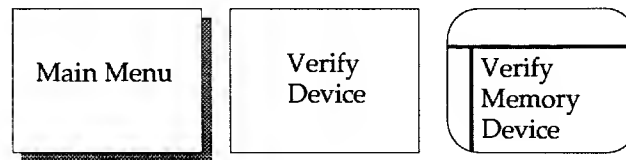
UniSite does not support vector testing for logic devices with more than 84 pins. Attempting to perform a vector test on a device with more than 84 pins will yield the following message:

```
OPERATION COMPLETE. Sumcheck = hhhhhhhh (Vector test
not supported)
```

- **Verify Passes (0,1,2)** — Selects the number of times to test the device. **0** directs UniSite not to test the device. **1** directs UniSite to test the device once at the device manufacturer's nominal Vcc. **2** directs UniSite to verify the device at the device manufacturer's recommended high and low Vcc levels. This parameter defaults to **2**.

### Verify Memory Device

If you select a memory device and then select the Verify command, UniSite's Verify Memory Device screen appears.



To verify a memory device, follow these steps:

1. Select and socket a memory device.
2. Specify the parameters you want. Then press **[J]** to begin the verifying.
3. When the verify operation is complete, UniSite displays the following message in the message bar:

```
OPERATION COMPLETE. Sumcheck = xxxxxxxx
```

The following parameters can be specified in the Verify Memory Device screen.

- **Source (R,D)** — Specifies the source of the data to use to verify the device. Press **Space** to toggle between R (RAM) and D (disk).
- **Filename** — Specifies the name of the disk file you want the data to be verified from. This option only appears if you have specified disk as the Source. The filename must follow standard DOS conventions, and can include a drive descriptor. An example of a valid filename is **27c256.dat**.
- **Data Word Width** — Sets the word width of the data to be verified. For 8-bit (or larger) devices, the minimum word width is equal to the device word width and the maximum is 64. For 4-bit devices, your word width choices are 4, 8, 16, and 32. This value should match the data bus word width in the target system for the device being programmed.
- **Next Device** — Designates the next device in the set. For example, if you are using 8-bit devices and have specified a word width of 16 bits on the Verify Memory Device screen, then two devices are required to verify each 16-bit word. Typing **1** for the next set member directs UniSite to verify the first device in the set with even-numbered addresses of the memory block. Typing **2** directs UniSite to use odd-numbered addresses.
- **Total Set Size** — Specifies how many virtual devices are in the set for device operations. For example, if you are verifying 16-bit wide data into two 8-bit wide devices, your virtual device still equals one (one 16-bit virtual device). Any number between 1 and 99 can be entered. Automatic Set Size calculation is attempted when any of the following parameters are changed: Device width, Device Block Size or User Data Size. The following equation defines Total Set Size:

$$\text{TOTAL SET SIZE} = \frac{\text{USER DATA SIZE}}{\text{DEVICE BLOCK SIZE} \left( \frac{\text{DATA WORD WIDTH}}{\text{DEVICE WIDTH}} \right)}$$

0987-1

- **User Data Size** — Specifies the hexadecimal size, in bytes, of the data block used to verify the device with the source. This value is normally equal to the device size or to a multiple of the device size for verifying a set. If 0 is entered, it is reset to the device size. User Data Size works with Total Set Size to determine the total amount of bytes to verify with a set of devices.
- **Next Operation Begins At** — Specifies the address in user memory where the next data byte will be verified. This value is calculated from the Data Word Width, Device Block Size, Memory Begin Address, device width, and next set member parameters.

**Optional Parameter Screens**

There are two types of parameter-entry screens: simple and complex. UniSite defaults to displaying the simple, Non-default Parameters screen. If you want to view the complex, All Parameters screen, press **[F4]** from the Non-default Parameters screen. If you have changed any of the default parameters, those changed parameters will also show up on the Non-default Parameters screen.

The following parameters are found on the complex, All Parameters Verify Memory Device screen.

- **Memory Begin Address** — Specifies the first address, in hex, of the first byte of data to be verified. For a SetSite operation, Memory Begin Address specifies the first address of the device in the first socket. If the Source is RAM, it is a beginning RAM address. If the Source is Disk, it is a beginning disk file address. The Memory Begin Address must be an even address if you have selected a 16-bit device. The default address is 0.
- **Device Begin Address** — Specifies the first hex device address that will be verified.
- **Device Block Size** — Specifies the size, in hex, of device data used in device operations. When you select a device, Device Block Size is automatically set to the device size and normally does not need to be changed. It is also automatically set to a smaller value if the Device Begin Address is nonzero. This parameter can be changed if desired. If a zero is entered, the Device Block Size is set automatically to equal the device size.
- **Set Auto-increment** — When enabled, directs UniSite (when in a set verify mode) to the starting memory address of the next data block that is to be verified. For example, if you have four 1K x 8 devices to verify against a 4K x 8 block of data, using the auto-increment option directs UniSite to point to the first address of the next 1K block after each device has been verified. For single device operations, this feature should be disabled and the Next Device parameter should be set to 1.

---

*Note: Items with an \* (asterisk) are visible on the screen only if the selected device supports the feature.*

- \* **Compare Electronic ID** — When enabled, compares the electronic signature of the device against the electronic signature of the selected algorithm.
- \* **Odd/Even Byte Swap (Y,N)** — When enabled, allows the Most Significant Bytes (MSB) and the Least Significant Bytes (LSB) of 16-bit words to be swapped when data is verified between a 16-bit device and memory. When disabled, data is verified with the MSB at an odd address. When enabled, the MSB is at an even address.
- **Verify Passes (0,1,2)** — Selects the number of times to test the device. 0 directs UniSite not to test the device. 1 directs UniSite to test the device once at the device manufacturer's nominal Vcc. 2 directs UniSite to verify the device at the device manufacturer's recommended high and low Vcc levels. This parameter defaults to 2.

---

## More Commands

In general, the commands found under the More Commands menu do things other than loading, programming, and verifying devices.

The More Commands is a multi-level menu with some commands nested three levels deep. The items on the top-level of the More Commands menu are described below:

- **Configure System** — Contains commands to perform the update operation, and edit, save, and restore UniSite's communications, interface, serial I/O, and programming parameters. (These are the items shown on the default parameters list at the beginning of this section.) From this menu, you can also select a new terminal type, and access Keep Current Configuration functions. You could use these commands to set up unique parameter files for each device type you want to program and then save those values from the More Commands/Configure System/Save screen. These parameter settings can then be loaded at a later time using the More Commands/Configure System/Restore screen.
- **Device Checks** — Performs device tests on socketed devices.
- **Edit Data** — Allows you to edit RAM or disk data. Separate editing features exist for logic and memory devices.
- **File Operations** — Performs various operations on UniSite's disk files, such as loading, saving, deleting, or renaming a file.
- **Job File** — Allows you to play back a series of keystrokes. This is useful if you are consistently programming the same devices. Up to ten job files may be stored on any Algorithm or System disk.
- **Remote Control** — Switches UniSite into remote mode, where it will accept commands sent from a remote computer. Chapter 6, "Computer Remote Control," lists the commands recognized by UniSite in remote mode.
- **Self-test** — Performs diagnostic checks on UniSite's circuitry.
- **Transfer Data** — Allows you to upload or download data to or from UniSite. Also allows you to select or change the data translation format.
- **Yield Tally** — Allows you to view or clear programming statistics.



## Configure System

The commands on the Configure System menu allow you to accomplish four basic tasks:

- Change communications protocols between UniSite and the other equipment connected to UniSite, such as a terminal, or a host computer.
- Configure the Remote and Terminal ports so they will be compatible with your terminal or host computer.
- Edit, save, or restore a set of programming features unique to the device type you want to program.
- Access Keep Current algorithm files.
- Access Custom Menu algorithm files.
- Update and change settings for a Mass Storage Module (MSM).

The commands available on the Configure Systems menu are described on the following pages.

## Edit

Use the commands on the Edit menu to change system parameters. These parameters include the settings of various options and features for the Programming, Serial I/O, Communication, and Interface screens.

Selecting Edit from the Configure System menu will display the Edit Parameter menu. From this menu you can choose Programming, Serial I/O, Communication, or Interface parameter screens which are described in the following subsections. Default parameter settings are shown in the table at the beginning of this chapter.

The system parameters that can be saved and restored with the More Commands/Configure System/Save and More Commands/Configure System/Restore commands are listed below.

**Programming Parameters**

Filename  
Source/Destination  
Reject Option  
Logic Verification  
Verify Passes  
Verify Data Format  
Data Word Width  
Algorithm Source  
User Data Size  
Memory Begin Address  
Device Begin Address  
Device Block Size  
High Speed Logic Drivers  
Illegal Bit Check  
Blank Check  
Compare Elec ID  
Enable Yield Tally  
Program Security Fuse  
Erase EE Device  
Odd/Even Byte Swap  
Continuity Check  
Compensated Vector Test  
RAM Device Selection

**Serial Port Parameters**

Baud Rate  
Parity  
Data Bits  
Stop Bits  
Enable CTS/DTR

**Communication Parameters**

Source/destination  
I/O Translation Format  
I/O Addr Offset  
I/O Timeout  
Upload Wait  
Transmit Pacing  
Download Echoing  
Output Record Size  
Number Of Nulls  
Instrument Control Code  
Fill Memory Option  
Fill Data  
High Speed Download  
User Menu Port \*  
JEDEC I/O Translate DIP/LCC Vectors  
Upload: Use End-of-file Delimiter  
Upload End-of-file Delimiter  
Download: Use End-of-file Delimiter  
Download End-of-file Delimiter  
Upload Host Command  
Download Host Command

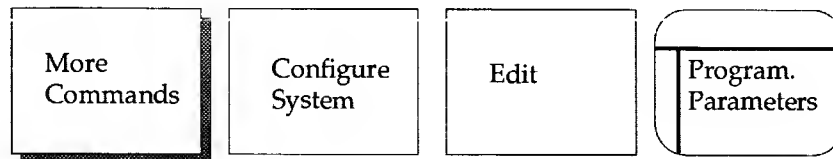
**Interface Parameters**

Power On CRC Mode  
Enable Terminal Beeps  
Remote On Code  
Remote Off Code  
Main Menu Job Files

\* This parameter can be saved with the Save System Configuration command and is read on power-up, but this parameter is not restored when a Restore System Configuration command is performed.

**Programming  
Parameters**

Use the Edit Programming Parameters screen to specify programming options, to enter memory block parameters, and to enable/disable different tests.



If you want to save the settings for use in a future programming session, then you should save them in a configuration file. Use the Save command (part of the Configure System menu) to save the settings for later use.

The different programming parameters and settings are explained below.

- **Filename** -- Specifies the disk file to use for load, program, and verify operations when the programming source is set to **D** (disk). The filename must follow standard DOS conventions, and can include a drive descriptor. An example of a valid filename is **a:16r8.dat**.
- **Source/Destination (R,D)** — Specifies the location of data to load, program, verify, download, upload or edit. Press **[Space]** to toggle between **R** (RAM) and **D** (Disk).
- **Security Fuse Data (0,1)** — To program the security fuse, set this parameter to **1** and set the Program Security Fuse parameter to **Y**. This parameter defaults to **0**, which disables programming of the security fuse.
- **Reject Option (C,S)** — Selects the number of times the device is pulsed with programming voltage before it is rejected as unprogrammable. **C** selects the number of pulses specified by the manufacturer. **S** selects either a one-pulse or the military-specification number of programming pulses. Unless you are programming devices to a strict military specification, you should leave this option set at **C**. This option defaults to **C**.
- **Logic Verification (F,V,A)** — Specifies the type of logic verification to perform. Select from Fuse Verification **F**, Vector Verification **V**, or All **A**. Fuse Verification checks the fuse pattern programmed into the device with the pattern in UniSite's memory. Vector Verification functionally tests the device, using structured test vectors stored in memory. **A** directs UniSite to perform both fuse verification and vector verification. Press **[Space]** to step through the three choices. This parameter defaults to **A**.

UniSite does not support vector testing for logic devices with more than 84 pins.

- **Verify Passes (0,1,2)** — Selects the number of times to test the device. 0 directs UniSite not to test the device. 1 directs UniSite to test the device once at the device manufacturer's nominal Vcc. 2 directs UniSite to verify the device at the device manufacturer's recommended high and low Vcc levels. This parameter defaults to 2.
- **Verify Data Format (B,H)** — Specifies either B (binary) or H (hex) for the mis-verify data display format on the Verify Memory Device screen.
- **Algorithm Type (D,E,K,C)** — Selects the type of device algorithms. D is the default and directs UniSite to select algorithms from the algorithms included on the Algorithm disk. If you set Algorithm Source to D and then choose the Select Device command, UniSite displays the standard Manufacturer List screen, as shown in Figure 4-4.

E directs UniSite to select algorithms from the alg.ext file. Extended algorithms (algorithms contained in the alg.ext file) are used by Data I/O to handle device approvals, special device algorithm updates, and the archived devices disk. If you set Algorithm Source to E and then choose the Select Device command, UniSite displays a Manufacturer List screen containing only the manufacturers found in the alg.ext file. If this parameter is set to E and UniSite cannot find the alg.ext file, UniSite displays the following message in the message bar:

Cannot access system file. Insert System disk.

K directs UniSite to select algorithms from the Keep Current algorithms it finds on the disks in the disk drives. If you set Algorithm Source to K and then choose the Select Device command, UniSite displays the Keep Current Part List screen, as shown in Figure 4-7. Keep Current algorithms are downloaded from Data I/O's Keep Current BBS and provide immediate support for new device algorithms and updated device algorithms. For more information, see the Keep Current documentation behind the Keep Current tab at the back of this binder.

C directs the programmer to select algorithms from the Custom Menu algorithms it finds on the disks in the disk drives. If you set Algorithm Source to C and then choose the Select Device command, the programmer displays the Custom Menu List screen.

---

*Note: See "Save System Parameters" on page 5-41 for instructions on how to save your Algorithm Type and other parameter changes as power-up defaults.*

- **Data Word Width** — Should match the data bus word width in the target microprocessor system for the device being programmed. For 8-bit (or larger) devices, any word width between 4 and 64 may be typed in. For 4-bit devices, your word width choices are 4, 8, 16, and 32. When performing a Quick Copy, the Data Word Width is set to the device word width and restored to the original value after the Quick Copy is complete.

With one exception, UniSite always changes this parameter to match the selected device's width. The exception is if the current Data Word Width is 16, your selected device's word width is 8 AND the previously-selected device's word width is also 8, this parameter does not change.

- **User Data Size** — Defines the hexadecimal size, in bytes, of the data block used in device operations. This value is normally equal to the device size or to a multiple of the device size for set programming. User Data Size works with Total Set Size to determine the total amount of bytes for a set operation. This parameter can also indicate the number of bytes in a data transfer operation.
- **Memory Begin Address** — Specifies the first hex address in RAM to load data from a device. Also specifies the first address where a device is programmed/verified. If the user memory is RAM, it is a beginning RAM address. If the user memory is disk, it is a beginning disk file address. The Memory Begin Address must be an even address if you have selected a 16-bit device. The default address is 0.
- **Device Begin Address** — Specifies the first address used in device operations. This option is used for memory devices only.
- **Device Block Size** — Defines the size, in hex, of device data used in device operations. When you select a device, Device Block Size is automatically set to the device size and normally does not need to be changed. It is also automatically set to a smaller value if the Device Begin Address is nonzero. This parameter can be changed if desired. If a zero is entered, the Device Block Size is set to the device size.
- **Illegal Bit Check (Y,N)** — Enables or disables the illegal-bit test. This test compares data in a device against data in UniSite's RAM to determine if the device has already-programmed locations of incorrect polarity. For example, UniSite returns an illegal-bit error in the following situation: data in RAM indicates a specific bit should be in an unprogrammed state while the corresponding bit in the device is in a programmed state. This parameter is enabled by default.
- **Blank Check (Y,N)** — Enables or disables the blank check test. A Blank Check test checks a device for programmed bits. This parameter defaults to Y, which enables the test.
- **Compare Elec ID (Y,N)** — Compares the electronic ID of the device against the electronic ID of the selected algorithm.
- **Enable Yield Tally (Y,N)** — When enabled, directs UniSite to keep a running tally of the programming yields for the last sixteen types of devices programmed. These totals show how many devices passed and failed, and what specific errors, if any, have occurred. This parameter defaults to N, which disables the test. A more complete description of this feature can be found in the "Yield Tally" section of this chapter.

- **Program Security Fuse (Y,N)** — Enables or disables the programming of the security fuse. To program the security fuse, set this parameter to Y and set the Security Fuse Data parameter to 1. This parameter defaults to N, which disables programming of the security fuse.
- **Erase EE Device (Y,N)** — Bulk erases the electronically erasable devices before UniSite attempts to program them.
- **Odd/Even Byte Swap (Y,N)** — When enabled, swaps data at memory address locations during a load, program, or verify operation. The contents of user RAM are not altered.

Swapping bytes is useful when manipulating 16- and 32-bit data for a target system that has a different architecture than the original file convention. For example, Motorola 16-bit data files store the Most Significant Bytes (MSB) at even-byte locations, and Motorola 32-bit data files store the significant bytes in descending order with the MSB in every first byte (byte 0) and the least significant in every fourth byte (byte 3). Intel 16-bit data files store the MSB at odd-byte locations and Intel 32-bit data files store MSM in ascending order with the most significant in the fourth byte.

The default for this parameter is N, which means the programmer maintains its RAM data and file data with the convention that the MSB of a 16-bit device resides in the odd byte of memory, and the MSB of a 32-bit word resides in the first of each four bytes of memory.

When set to Y, for a 16-bit device, data is loaded/programmed/verified into user memory with the MSB at even addresses. For 32-bit devices, the significant bytes are placed in ascending order with the MSB placed in user memory in the fourth of each four bytes (the fourth byte is swapped with the first byte and the third byte is swapped with the second byte).

- **Continuity Check (Y,N)** — Checks for open device pins before programming a device. This parameter is enabled by default, and also when a new device type is selected.
- **Serial Vector Test (Y,N)** — When enabled, this test applies each vector's input states serially, starting with pin one and stepping through the remaining pins. This test is a diagnostic tool designed to help debug and classify test vector failures. Specifically, this test is designed to isolate test vectors that are sequence dependent. If a sequence-dependent vector is found, it should be broken into two or more vectors to make them sequence independent.

The JEDEC specification for test vectors requires that test vectors be sequence independent. If sequencing between pins is important, then the test vector should be separated into two or more vectors to make them sequence independent. This test helps isolate vectors that are sequence-dependent and should be expanded.

This switch is available only for logic devices and is disabled on power up. The switch is also returned to N when parameters are restored or when another device is selected.

- **High Speed Logic Drivers (Y,N)** — When enabled, this feature increases the speed of the logic transitions between 0 to 1 and 1 to 0 of the test vector input states.

The speed of the logic transitions is increased by driving the 0 and 1 levels using the high speed logic drivers instead of a current limited driver.

The JEDEC specification for test vector 0 and 1 input states defines that these inputs be current limited, so that the outputs of the device under test can overdrive the 0 or 1 level. However, current-limited drivers have inherently slow transition speeds. Enabling this feature reduces the possibility of doubling clocking due to slow transition times.

This switch is available only for logic devices and defaults to Y on power up and stays on until turned off.

---

**CAUTION:** *If used with invalid test vectors that drive outputs, the High Speed Logic Drivers test may cause over current errors.*

- **Compensated Vector Test (Y,N)** — Y enables load compensation on PLD output pins under test during vector testing. This may eliminate structured test errors when testing PLDs sensitive to output loading, where many of the device's registers transition simultaneously. This test is available only for logic devices and defaults to N at power up. This parameter defaults to Y if you select a non Open Collector device and defaults to N if you select an Open Collector device. This parameter can be saved/restored with the Save/Restore Configuration command.

- **RAM Device Selection** — When selected, allows the device programming algorithms to reside in user RAM. This feature requires UniSite to contain the 4 MB or 8 MB option and allows much faster device selection than the normal method of loading the algorithm from the disk each time a device is selected.

After powerup, when you enable RAM Device Selection and select a device, UniSite loads the algorithms from the algorithm disk(s) you have inserted. The following message appears:

```
Loading device algorithm file into user RAM
```

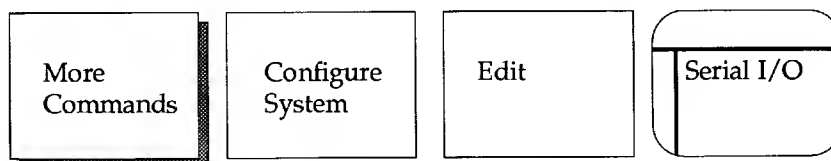
If only one algorithm disk was present when you enabled RAM Device Selection, the first time you select a device whose algorithm is not already loaded into RAM, you will be prompted to insert the other algorithm disk, as follows:

```
Cannot access system file. Insert Algorithm Disk X
```

Similarly, if RAM selection is enabled when the programmer is powered up, the algorithms from the disk in drive B are loaded into RAM. The first time you select a device that requires algorithms from the other algorithm disk, UniSite prompts you to insert the other disk so that its data can be loaded into RAM.

## Serial I/O Port Configuration

Use the Edit Serial I/O Port Configuration command to specify the communications parameters for UniSite's two serial ports.



Use this command when connecting equipment to UniSite's Terminal and Remote ports so they can be compatible with your terminal or host computer. If you want to save the settings of the two ports for use in a future session, you should save them in a configuration file.

The different parameters and their settings are explained below.

A change in the serial port parameters does not become effective until you press ☐. If terminal settings are changed, a message appears, prompting you for another ☐ after you have altered your terminal to match the new settings. Output to the terminal is suspended until you enter the second ☐.

- **Parity (N,O,E)** — Three options are available for the parity setting: N (No parity), O (Odd parity), and E (Even parity). Press ☐ to cycle through the three values.
- **Data Bits (7,8)** — Specifies the number of data bits UniSite recognizes during serial communication. Press ☐ to toggle between the two values.
- **Stop Bits (1,2)** — Specifies the number of stop bits between data bytes. Two stop bits are generally used for baud rates of 110 or lower. Press ☐ to toggle between the two values.
- **Enable CTS/DTR (Y,N)** — Enables or disables CTS/DTR hardware handshaking. Press ☐ to toggle between the two values.
- **Baud Rate** — Specifies the baud rate for both the Terminal and Remote ports. Press ☐ to cycle through the baud rates supported by UniSite.

UniSite supports the following baud rates: 50, 75, 110, 134.5, 150, 200, 300, 600, 1050, 1200, 1800, 2000, 2400, 4800, 7200, 9600, and 19.2 kbaud (and 115.2 kbaud with HiTerm). However, not every baud rate works on one port when certain baud rates are selected for the other port. If you select incompatible baud rates, UniSite beeps and displays the following error message:

WARNING: Selection not compatible with other channel!

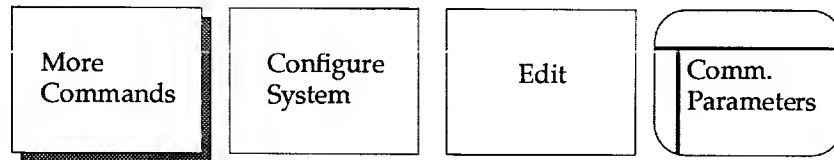
If one of the rates on the left is used for either port, then the corresponding rate on the right does not work on the other port.

50	75, 150
150	200, 1050
7200	75, 150, 1800, 2000, 19.2K
19.2K	50, 200, 1050, 7200



**Communication Parameters**

Use the Edit Communication Parameters screen to change UniSite's I/O characteristics when downloading or uploading data.



If you want to save the settings of the communication parameters for use in a future session, you should save them in a configuration file. Use the Save command (part of the Configure System menu) to save the settings for later use.

- **Source/destination (R,T)** — Specifies the source/destination port of the data to be transferred. Press **Space** to toggle between R (Remote port) and T (Terminal port).
- **I/O Translation Format** — Specifies the two digit decimal code that corresponds to the translation format in which the data will be transferred. For a detailed list and sample of each format, see the Translation Formats chapter of this manual.
- **I/O Addr Offset** — Specifies address offset to use during a data transfer. The I/O Addr Offset is subtracted from the I/O Addresses during Input from Disk and Download operations, so data is properly located in User RAM or on disk. During Output to Disk and Upload operations, the I/O Addr Offset is added to the User Memory Address.

Specifying FFFFFFFF instructs UniSite to set the I/O Addr Offset to the first incoming address of data received from Input from Disk and Download operations. During Output to Disk and Upload operations, setting the I/O Addr Offset to FFFFFFFF is the same as setting the I/O Addr Offset to 0, since no offset is added to the I/O addresses sent out.

To load your file absolutely, set the I/O Addr Offset to 0; the data is stored at the addresses specified in the data file.

This parameter defaults to FFFFFFFF.

- **I/O Timeout** — Limits the time, from 0 to 99 seconds, UniSite waits for a data transfer to begin. A value of 0 disables the timeout. A value of 0 should not be used for data formats which don't use a start code and an end code because these formats need the timeout enabled in order to terminate a normal data transfer. Formats which fall into this category include codes 5, 6, 7, 9, 11, 13, 16 and 96.
- **Upload Wait** — Sets the period that UniSite waits before it begins sending data to the host computer after the host upload command is sent. The range of this parameter is 0 to 99 seconds.
- **Transmit Pacing** — Specifies the time-delay to insert between characters transmitted to a host during an upload. See the note about transmit pacing at the end of this section for more information.

- **Download Echoing (Y,N)** — Displays the data being downloaded. This may slow down UniSite in receiving data and is not recommended for high baud rates, such as 9600 and above. Download echoing may not be used with the binary formats.
- **Output Record Size** — Specifies the number of data bytes contained in each data record during upload. The range of this parameter is 0 to 256 bytes. Some formats have fixed record lengths for which this parameter does not apply.
- **Number Of Nulls** — Sets the number of null characters sent between each record in a data file after a carriage return and line feed. The range of this parameter is 0 to 254 nulls. Entering 255 specifies no nulls and suppresses the line feed.
- **Instrument Control Code** — Specifies how the data transfer will be controlled. Selecting 0 specifies regular XON/XOFF handshaking, selecting 1 or 2 specifies a special handshaking sequence (see the Translation Formats chapter for more information).
- **Fill Memory Option (N,D,U)** — Specifies what data user memory will be filled with before downloading begins. This option is used for Download Data and Input from Disk operations. If you select N (none), user memory will not be changed and whatever is in user memory will be overwritten by the downloaded data. If you select D (default), user memory will be filled with the data appropriate to initialize unused locations to the unprogrammed state for the device type selected. If you select U (user defined), then user memory will be filled with whatever you specify in the Fill Data option.
- **Fill Data** — Allows you to type in the hexadecimal data to be placed at unused locations of user memory during download. To use this option, you must also specify U for the Fill Memory option. User memory can also be filled with the specified data during the Edit Data operation.
- **High Speed Download (Y,N)** — When enabled, this parameter allows UniSite to download data from a PC at 115.2 kbaud. For high speed download to work, this parameter must be set to Y and the following conditions must be met:
  - You must have an IBM-compatible PC connected to UniSite's Remote port.
  - You must have Data I/O's HiTerm software installed and running on the PC.
  - The data you are downloading must be stored in a data format supported by HiTerm for high speed downloads. The HiTerm manual contains a list of supported formats.
  - You must use HiTerm's TR command when you download data to UniSite. See the HiTerm manual for more information.

Setting this parameter to Y causes the User Menu Port parameter to be set to R (Remote port). See the "Setting up High Speed Download" section of Chapter 2 for more information.

- **User Menu Port (R,T)** — Specifies which of UniSite's two ports should be used to send user menu information and to receive commands. (User menu information and commands are what you see when you operate UniSite from a workstation or a terminal.) Normally, this parameter is set to T (Terminal port).

If you want to use the 115.2 kbaud high-speed download option, use this parameter to redirect the user menu information from the Terminal port to the Remote port. To redirect the user menu information, set this parameter to the port you want connected to the controlling PC and press ☐. Then move the cable to the port specified on UniSite.

---

*Note: When switching ports, make sure the communication settings of the terminal/PC and the port you switch to are the same. Also, make sure you switch between compatible terminals; for example, you cannot switch from an ANSI 3.64 compatible terminal to a VT-100 compatible terminal.*

Setting the High Speed Download parameter to Y causes this parameter to be set to R (Remote port).

- **JEDEC I/O Translate DIP/LCC Vectors (Y,N)** — When enabled, translates test vectors for a device from its DIP package to its PLCC/LCC package. If this feature is selected, UniSite alters the test vectors during I/O translation, allowing for the different pinouts of the two package types. During downloading, vectors are converted from DIP to PLCC/LCC; during uploading, vectors are converted from PLCC/LCC to DIP. Use this feature if you have created test vectors for a DIP device but actually want to program the PLCC/LCC version of the same device.
- **Upload: Use End-of-file Delimiter (Y,N)** — When enabled, inserts an end-of-file character following an uploaded file. The delimiter character signals the host system that the upload is complete. During an upload, an end-of-file character is transmitted to the host. To invoke this feature, you must select this option and provide the two-digit hexadecimal number of the ASCII character you want to use as the end-of-file character. Select any value between 01 and 1F.

---

*Note: Use this feature only if you are using a format with an end-of-text character. It cannot, for example, be used for files stored in a binary data translation format.*

- **Upload End-of-file Delimiter (1-1F)** — Selects the two-digit hexadecimal number of the ASCII character you want to use as the end-of-file character in uploading data to a host computer. Select any value between 01 and 1F.
- **Download: Use End-of-file Delimiter (Y,N)** — When enabled, signals UniSite that the transmission of a particular file is complete. During a download operation, characters after the last record and before the end of the file would be ignored. If you wish to use this feature, you must select this option and you must provide the two-digit hexadecimal number of the ASCII character you want to use as the end-of-file character. Any value between 01 and 1F may be selected.

- **Download End-of-file Delimiter (1-1F)** — Selects the two-digit hexadecimal number of the ASCII character you want to use as the end-of-file character in downloading from a host computer. Any value between 01 and 1F may be selected.
- **Upload Host Command** — Type the command you want to use to tell the host system what to do with the data to be uploaded. The command may be up to 58 characters long. You can use host operating system commands on this line. For example, with UNIX, you would enter an upload command such as `cat > 27128.hex`. UniSite appends a `␣` to the command.
- **Download Host Command** — Type the command you want to send to the host system to initiate a file transfer download to UniSite. The command may be up to 58 characters long. You can use host operating system commands on this line. For example, with UNIX, you would have a download command such as `cat 27128.hex`.

### Transmit Pacing

Transmit pacing provides a time delay between characters sent to the host by UniSite, affecting data uploaded from UniSite as well as host commands sent by UniSite. If you encounter errors during upload operations to a host, you must use the transmit pacing delay feature to eliminate the errors. This will provide a character-by-character delay to prevent data overrun on the host. This type of condition may exist in spite of the use of hardware and/or software handshaking on the host and UniSite. This condition is most likely to occur on hosts which cannot accept incoming data fast enough at high baud rates.

Since the host may not explicitly report any error, to determine if errors are occurring during an upload process, transfer data to the host by using the upload function with an I/O format selected which utilizes checksums (such as Format 87). Note the checksum reported by UniSite when the transfer is complete. Then transfer the same data back to UniSite while UniSite is performing the compare data function, again noting the checksums. If the checksums don't match or if an error is encountered, a transmit pacing delay should be used.

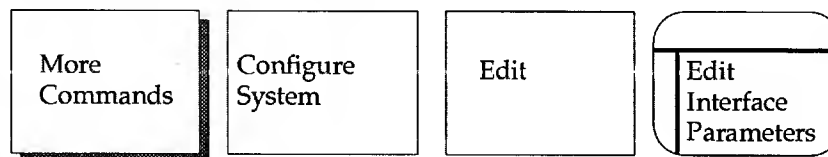
Typical delay values are presented in the following table. The transmit pacing value is specified in tenths of milliseconds delay. For example, a value of 12 represents 1.2 milliseconds delay. The minimum delay possible (other than zero) is 4 (0.4 milliseconds) and the maximum is 99 (9.9 milliseconds). The factory default is 0. The transmit pacing value required for reliable data transfers may vary somewhat from those presented in the table due to the particular characteristics of the host involved. This is primarily determined by the processor speed of the host and whether or not software other than the communication software is running at the same time on the host (in the latter case larger delay values may be required).

### Transmit Pacing Delay Values

	4800 and less	9600	19.2K
HiTerm			
PC	0	0	6
AT	0	0	6
VTERM			
PC	0	4	15
AT	0	0	8
PROCOMM			
PC	0	4	9
AT	0	0	6

### Interface Parameters

Interface Parameters are parameters that are not related directly to uploading/downloading of files.

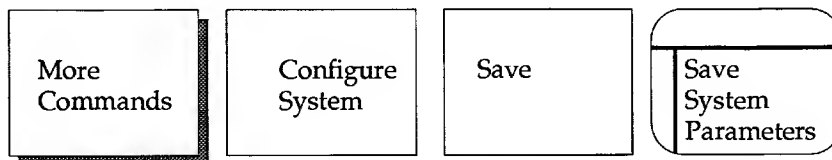


The parameters on the interface parameters screen are described below.

- **Power On CRC Mode (Y,N)** — When enabled, this feature allows you to power up UniSite and have it automatically enter computer remote control mode. You can exit CRC by pressing **[Ctrl] + [Z]**.
- **Enable Terminal Beeps (Y,N)** — When enabled, UniSite beeps each time an error message is generated.
- **Remote On Code** — When enabled, you can use an ASCII character to enable the Remote port. Type the two-digit hexadecimal number that represents the ASCII character you want to use to enable remote control.
- **Remote Off Code** — When enabled, you can use an ASCII character to disable the Remote port. Type the two-digit hexadecimal number that represents the ASCII character you want to use to disable remote control.
- **Main Menu Job Files (Y,N)** — When enabled, you can start a Job File from the Main Menu rather than having to go to the Job Files menu.

## Save System Parameters

System Parameters are all the parameters on the Programming, Serial I/O, Communication, and Interface screens. With the Save command, you can save a set of system parameters for future use.



This feature is useful if you want UniSite to power up with preset parameters. This feature is also useful if multiple users prefer to have their own set of parameters easily available.

To save a system configuration, follow these steps:

1. Before using the Save command, go through the above-mentioned Configure/Edit menus, setting the system parameters to fit your particular application.
2. When you have UniSite configured the way you want, go to the Save screen and select a file number in which to store the configuration file. The file number must be between two and nine. File numbers zero through two are reserved for factory defaults, power-up defaults, and CRC defaults.
3. Next, enter a description of the configuration file you are saving, for example, **Config file for Intel 27C256**. The name can be up to 30 characters long. After you enter the file description, press . UniSite displays the following message in the message bar:

Parameter Entered

4. Finally, to save the current system configuration to a configuration file, press . While saving the configuration, the action symbol rotates. When done, UniSite displays the following message:

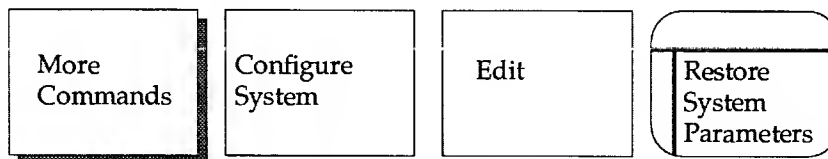
System parameters saved

---

*Note: If a device has been selected, it is saved as part of the configuration file.*

## Restore System Parameters

System Parameters are all the parameters on the Programming, Serial I/O, Communication, and Interface screens.



With the Restore command, you can restore a set of previously saved system parameters.

To restore a system configuration, follow these steps:

1. Select Restore from the Configure System Parameters menu.
2. UniSite displays a list of all the configuration files that have been saved on the System disk. Look at the list of configuration files and find the file number of the file you want to restore.
3. Enter the file number of the configuration file you want to restore and press . UniSite loads the system parameters and displays the following message in the message bar when done:

System parameters restored

---

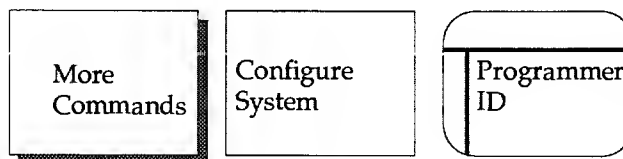
*Note: When you restore a configuration, the device algorithm selected when the configuration file was saved (if any) will also be restored.*

*If you are restoring a configuration that had a device selected as part of the configuration and UniSite cannot find the algorithm for the selected device, UniSite will prompt you to insert another Algorithm disk.*

When you return to a screen, the cursor will be where you left it. Restoring parameters to factory defaults or power-up defaults returns the cursor to its original position. Restoring parameters to user-defined defaults has no effect on cursor positions.

## Programmer ID

Use this command to view your programmer ID.



## Carrying a Configuration File Forward

Beginning with version 4.10 you can carry system configurations forward after you update to a new version of system software. Previously, you had to recreate system configurations when you updated to a new version of system software.

---

*Note: If you carry the system configurations forward, the power up defaults will be carried forward in addition to any other user-defined system configurations. This is because the power-up defaults, in addition to the factory defaults and any user-defined configurations, are stored in a file named sysparm.sys on the UniSite System disk.*

Now, the following scenario is possible:

- Under version X.3, create and save two different system configurations, such as one for memory devices and one for logic devices.
- Update to version X.4 system software.
- Carry the system configurations created under version X.3 forward to X.4.
- Use the two system configurations with version X.4.

---

*Note: This procedure works only with system software version 4.10 and later.*

For more information on saving and restoring system configurations, and on system configurations in general, see the "Restore System Parameters" and the "Save System Parameters" sections in this chapter.

## Floppy or MSM

The procedure described in this section can be used to carry configuration files forward from an old System disk to the new System disk. If you wish to carry configuration files that are located on a Mass Storage Module (MSM) forward, refer to "Install and Update the MSM" on page 2-32.

## What You Need

To carry system parameter settings forward from the old System disk to the new System disk, you will need a DOS-based IBM-compatible PC with a 3.5-inch floppy drive capable of reading and writing 720KB floppy disks.

## Carrying the Configuration File Forward

Follow the steps below to carry your system configurations forward to a new version of system software.

---

*Note: In the following steps "old" refers to the System disk containing the system configurations you want to carry forward to the "new" System disk.*

1. Power up your programmer using the new System disk.
2. Make a backup copy of the new System and Algorithm disks. See the description of the Duplicate Disk command for more information on making a backup copy of your System and Algorithm disks.



---

*Note: Store the master copies in a safe place. Use your backup copies for day-to-day operation.*

3. Power up the PC. When the PC has finished booting, insert your working copy of the old System disk in the 3.5-inch drive in the PC.

---

*Note: Make sure you insert the System disk with the configurations you want to carry forward.*

4. Copy the sysparm.sys file to your hard drive. Enter the command  
**copy drive1:\sysparm.sys drive2:**

where *drive1*: is the drive designator for the 3.5-inch disk drive and *drive2*: is the drive on which you will be temporarily storing the copy of the sysparm.sys file.

---

*Note: We suggest you work with the backup copies of your System and Algorithm disks and not with the originals.*

5. Remove the old System disk and insert the new System disk in the 3.5-inch drive. Make sure the new System disk is write enabled.

Copy the sysparm.sys file to the new System disk. Enter the command

**copy drive2:sysparm.sys drive1:\sysparm.sys**

where *drive2*: is the drive you copied the sysparm.sys file to in step 4.

6. If you have more UniSites that you want to copy a configuration file to, repeat steps 5 for each of the remaining UniSites. For example, if you have three UniSites in a lab, and if you used the same configuration files on all three, the following scenario would be possible:

- Update all three UniSites to a new version of system software.
- Make backups of the new System and Algorithm disks.
- Perform steps 1 to 5 as described above on one UniSite.
- Repeat step 5 for the remaining two UniSites.

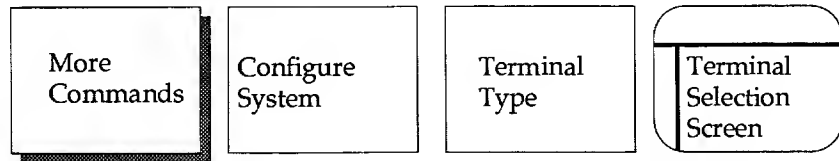
7. Remove the disk from the 3.5-inch drive insert a new System disk into UniSite. In the More/Configure system/Restore screen, press **[F4]** to reconstruct the parameter list.

---

*Note: If you do not reconstruct the parameter list, the next time your programmer is powered up with the new System disk, it prompts System parameter field is out of date. Do you want to update it (Y/N). Type Y and press **[Enter]** to reconstruct the parameter list.*

## Terminal Type

This command changes the current and default terminal types.



A list of compatible terminal types follows the instructions for changing the terminal type.

To change the current terminal type or default terminal type, follow these steps:

1. If you have not already done so, configure your terminal to match one of the compatible terminal types.
2. Select the Serial I/O Parameters screen from the More Commands/Configure System/Edit/Serial I/O menu and observe the settings of the port you want to connect the new terminal to. If the terminal's communications protocol does not match the port's, change the settings of the new terminal to match the port's settings.
3. Select the More Commands/Configure System/Terminal Type command. At this point UniSite displays the default and current terminal types, and the available terminal types. Select a terminal type, enter the number corresponding to that terminal type and press ☐. You have changed the terminal type for this current session.
4. UniSite then prompts you with the following:  

```
Save terminal type as power on default? (Y/N) [N]
```

If you want to change the default terminal type, then continue with step 5. If you do not want to change the default terminal type, then go to step 6.
5. If you do not want to change the default terminal, press ☐ ☐ and go to step 7.
6. To change the default terminal type, press ☐ ☐. UniSite saves the new terminal type to disk. The new terminal type is now part of the power-on parameters.
7. The screen clears and UniSite returns to the Configure System Parameters menu. Resume normal operation.

## Approved Terminals

UniSite is compatible with the terminal types listed below. If your terminal is not included in the list, refer to the manual supplied with the terminal to determine if it can emulate one of those listed below.

- ANSI 3.64 compatible terminals
- DEC VT-100 compatible terminals
- Qume QVT-101 compatible terminals
- TELEVIDEO TVI-910 compatible terminals
- Wyse WY-30 compatible terminals

## Keep Current

The commands in the Keep Current menu allow you to access Keep Current algorithm files (.KCx). Commands available on this menu include the following:

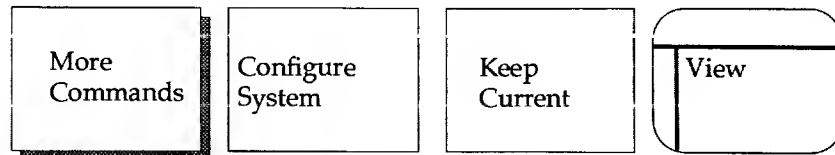
- View
- Replace/Restore
- Delete
- Purge

---

*Note: For information on how to select a device supported by a Keep Current algorithm, refer to "Select Device" on page 5-5.*

### View

This command allows you to view a list of .KCx files.



View displays information on all .KCx files found on both drives A and B. Compatibility between system software and Keep Current algorithms is not checked.

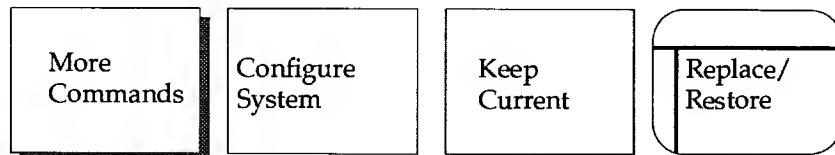
To view Keep Current algorithm files, follow these steps:

1. Insert the disk with the .KCx files you want to view into either disk drive.
2. When you select the View command, the dialog window fills with a directory listing. UniSite displays up to 10 files at one time. If there are more than 10 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the first page of files.

If you want to view files on another disk, press **F2**, insert another disk, and return to the beginning of this step.

### Replace/Restore

This command displays the Replace/Restore screen. In the Replace/Restore screen, Keep Current algorithms can be toggled between "replaced" and "restored" status.



If a part is marked as "replaced," the Keep Current algorithm is used instead of its corresponding algorithm in the default algorithm sets during normal device selection. Parts previously marked as "replaced" can be restored. In this case, the default algorithm is used during normal device selection for that part.

To toggle algorithm(s) between "replaced" and "restored" status, follow these steps:

1. Insert both the System disk (this has the **kcmarker.sys** file on it) and the disk with the .KCx files you want to replace or restore into the disk drives.
2. When you select the Replace/Restore command, UniSite loads the **kcmarker.sys** file and checks to see if **mem\_alg.sys** or **log\_mem.sys** have been loaded into RAM. If the file it needs has not been loaded, UniSite searches for the algorithm disk. If the algorithm disk is not found, the following message appears:

Cannot access system file. Insert algorithm disk.

If this message is displayed, insert the appropriate algorithm disk, and try again.

If no Keep Current algorithms are found or the algorithms are not compatible with the current version of your system software, the following message is displayed:

Insert Keep Current algorithm disk

If this message is displayed, insert a disk with compatible Keep Current algorithms into one of the disk drives, and try again.

3. On the Replace/Restore screen, the dialog window fills with a directory listing with parts marked as "replaced" displayed in reverse video. UniSite displays up to 10 files at one time. If there are more than 10 files, press **[Ctrl] + [N]** to display the next page of files. Press **[Ctrl] + [P]** to display the first page of files.

Note that not all .KCx files are displayed. The only files that are displayed are those that

- Have a corresponding algorithm on one of the Algorithm disks (the algorithm can already be selected during the normal device selection)
- Are compatible with your version of the system software

If you want to view files on another disk, press **[F2]**, insert another disk, and return to the beginning of this step.

4. Move the cursor to the Replace/Restore field and enter the number corresponding to the file you want to replace or restore.
5. To toggle the file, press **[J]**. If you do not want to toggle the file, press **[F2]** to return to the Keep Current Configuration menu.

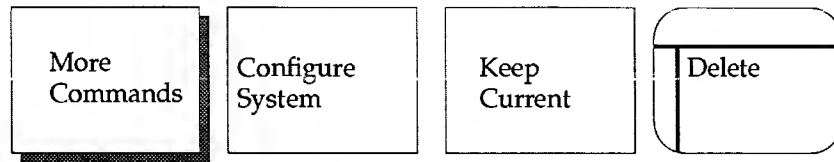
If you toggle the part to "replaced" status, it is displayed in reverse video. If the part was already marked as "replaced," it is toggled to "restored" status.

---

*Note: The maximum number of replaced algorithms is 10.*

## Delete

This command deletes a .KCx file from a disk.



To delete a file from a disk, follow these steps:

1. Insert the disk with the .KCx file you want to delete into either disk drive.
2. When you select the Delete command, the dialog window fills with a directory listing. UniSite displays up to 10 files at one time. If there are more than 10 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the first page of files.

If you do not see the file you want to delete, press **F2**, insert another disk, and return to the beginning of this step.

3. Move the cursor to the Delete field and enter the number corresponding to the file you want to delete.
4. Move the cursor to the Are you sure field and press **Y**.

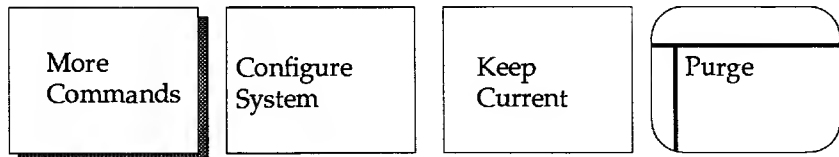
---

**CAUTION:** *If you do not want to delete the file, do not press Enter.*

5. To delete the file, press **↓**. If you do not want to delete the file, press **F2** to return to the Keep Current Configuration menu.

**Purge**

This command deletes all outdated .KCx files from a disk leaving only the most up-to-date algorithms.



To purge files from a disk, follow these steps:

1. Insert the disk with the .KCx files you want to purge into either disk drive.
2. When you select the Purge command, the dialog window fills with a directory listing of outdated Keep Current files found on both drives A and B (outdated Keep Current algorithm files have version numbers older than the current system software version number). UniSite displays up to 10 files at one time. If there are more than 10 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the first page of files.

If you do not see the files you want to purge, press **F2**, insert another disk, and return to the beginning of this step.

3. In the Are you sure field?, press **Y**.

---

**CAUTION:** *If you do not want to purge files, do not press Enter.*

4. To purge files displayed on the screen, press **Enter**. If you do not want to purge files, press **F2** to return to the Keep Current Configuration Operations menu.

If no more .KCx files are left on the disk(s), UniSite will return to the Keep Current Configuration menu.

## Custom Menu Algs

The commands in the Custom Menu Algs menu allow you to build Custom Menus. You can select devices from a Custom Menu.

The most common use of a Custom Menu is to create a list of the devices you most commonly use. Thus, when you wish to select a device, instead of scrolling through screens of devices that you rarely use, you can choose from a smaller, custom list of those devices you most often use.

The commands available on this menu include the following:

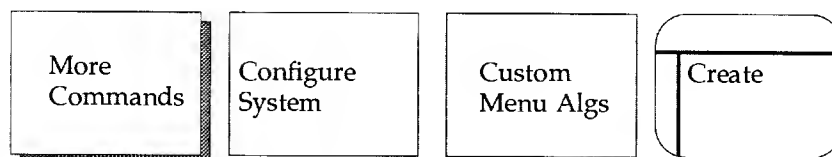
- Create
- Add
- View
- Delete
- Update

---

*Note: For information on how to select a device from a Custom Menu, refer to "Select Device" on page 5-5.*

### Create

This command displays the Create screen. In the Create screen, you can create a new Custom Menu by selecting the devices you wish to include in the Custom Menu. The Create command also copies device related files to the new Custom Menu disk (the disk that will contain the Custom Menu).



To create a new Custom Menu, follow these steps:

1. Select Create. The programmer prompts you for

Algorithm media (F,M) — Only appears if MSM installed  
Algorithm source (D,E,K)  
Custom Menu algorithm disk

where (F, M) is floppy disk or Mass Storage Module (MSM) and (D, E, K) is default algorithms, extended algorithms, or Keep Current algorithms.

2. Select the **algorithm media** and **algorithm source** that matches the first algorithm you wish to add to the Custom Menu. (You are not limited by the algorithm source you choose. Your Custom Menu can contain any combination of default, extended, and Keep Current algorithms.) In the **Custom Menu algorithm disk** field, select the disk drive where the Custom Menu algorithm files will be created.
3. If prompted, insert the Algorithm disk that contains the algorithms from which you want to choose into the disk drive. (If you do not have the correct disk inserted, the programmer will prompt you for the disk to insert when necessary.)

4. Press **[J]** to continue. Follow the directions on the screen (inserting disks if prompted).

---

**CAUTION:** *This operation uses RAM as a temporary storage buffer and alters the contents of RAM.*

If you get the message

Need to clear user RAM file(s) prior to operation...

this means that the programmer does not have enough available RAM to perform the operation. If you press **[J]**, the RAM file(s) are cleared and the operation continues. If you don't want RAM files cleared, cancel the operation by pressing **[Ctrl] + [Z]**.

5. When the programmer is ready to write the Custom Menu (CM) algorithm list to a disk, the following message is displayed:

Insert Custom Menu algorithm disk...

Insert the disk that you want the Custom Menu files to be written to and press **[J]**.

If you get the message File ERROR: Cannot allocate file space, you need to delete some files on your target disk to make more room for the CM files or insert a different disk that has enough space for the files.

6. After the programmer is done copying files, the programmer displays a list of devices (if algorithm type K was selected in step 2) or a list of device manufacturers (if algorithm type D or E was selected in step 2).

Examine the list and select a device or device manufacturer by typing the number corresponding to the selection and pressing **[J]**. If the device or device manufacturer you want is not listed, type **[Ctrl] + [N]** to see the next page of devices or manufacturers.

7. If you selected a manufacturer in step 6, a list of devices produced by the selected manufacturer appears. Type the number corresponding to the device you wish to add to your Custom Menu, and press **[J]** to add the device. (**[Ctrl] + [N]** to see the next page of devices.)
8. If prompted, insert the Algorithm disk that contains the algorithm you wish to add. (If you do not have the correct disk inserted, the programmer will prompt you for the disk to insert when it is necessary.)

If you get the message Cannot access file\_name.sys the programmer could not find a file on the disk inserted into the programmer's drive. Insert the Algorithm disk that contains the file.

9. Select additional devices from the current screen (or press **[F2]** to return to the manufacturer list if you were selecting devices from a manufacturers list). Repeat steps 6 through 8 as necessary.

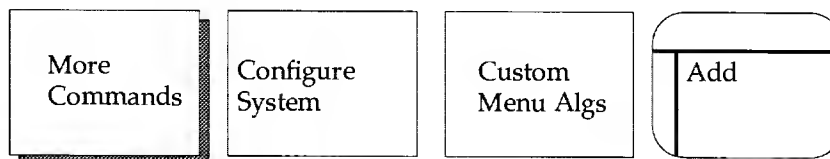
When you are done adding devices, press **[F2]** twice until you are prompted to insert the Custom Menu disk.



10. Insert the Custom Menu disk and press ☐ to save your changes to disk. To add additional devices to your Custom Menu, use the Add command (described below).

## Add

This command displays the Add screen. In the Add screen, you can add new devices to a Custom Menu.



---

*Note: Only use the Add command to add devices to an already created Custom Menu (a Custom Menu created using the Create or Update command).*

To add devices to a Custom Menu, follow these steps:

1. Select **Add**. The programmer prompts you for the  
Algorithm media (F,M) – Only appears if MSM installed  
Algorithm source (D,E,K)  
Custom Menu algorithm disk  
  
where (F, M) is floppy disk or Mass Storage Module (MSM) and (D, E, K) is default algorithms, extended algorithms, or Keep Current algorithms.
2. Select the **algorithm media** and **algorithm source** of the device you wish to add to your Custom Menu. In the **Custom Menu algorithm disk** field, select the disk drive where the Custom Menu is located.
3. If prompted, insert the Algorithm disk that contains the algorithms from which you want to choose into the disk drive, and press ☐ to continue. Follow the directions on the screen (inserting disks if prompted).

---

**CAUTION:** *This operation uses RAM as a temporary storage buffer and alters the contents of RAM.*

4. The programmer displays a list of devices (if algorithm type K was selected in step 2) or a list of device manufacturers (if algorithm type D or E was selected in step 2).  
  
Examine the list and select a device or device manufacturer by typing the number corresponding to the selection and pressing ☐. If the device or device manufacturer you want is not listed, type **[Ctrl] + [N]** to see the next page of devices or manufacturers.
5. If you selected a manufacturer in step 4, a list of devices produced by the selected manufacturer appears. Type the number corresponding to the device you wish to add to your Custom Menu, and press ☐ to add the device. (**[Ctrl] + [N]** to see the next page of devices.)

6. If prompted, insert the Algorithm disk that contains the algorithm you wish to add. (If you do not have the correct disk inserted, the programmer will prompt you for the disk to insert when it is necessary.)

If you get the message `Cannot access file_name.sys` the programmer could not find a file on the disk inserted into the programmer's drive. Insert the Algorithm disk that contains the file.

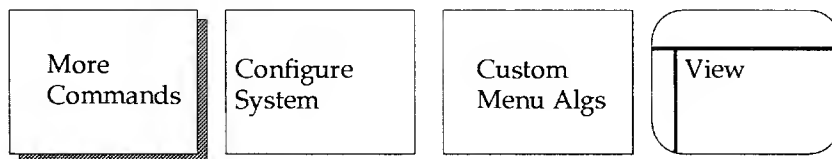
7. Select additional devices from the current screen (or press **F2** to return to the manufacturer list if you were selecting devices from a manufacturers list) or press **F2** (once or more if necessary) to return to Add screen in order to select a different algorithm media or algorithm source. Repeat steps 2 through 6 as necessary.

When you are done adding devices, press **F2** twice until you are prompted to insert the Custom Menu disk.

8. Insert the Custom Menu disk and press **↓** to save your changes to disk.

## View

This command displays the contents of a Custom Menu.



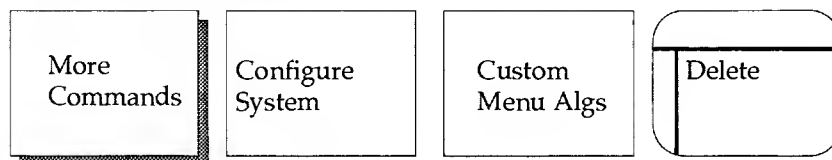
To view a Custom Menu, follow these steps:

1. Select the **View** command.
2. Insert the disk that contains your Custom Menu and press **↓** to continue. Follow the directions on the screen (inserting disks if prompted).

When you are done viewing the Custom Menu, press **F2** to return to the Custom Menu Algs menu.

## Delete

This command displays the Delete screen, which allows you to delete devices from a Custom Menu.



To delete devices from a Custom Menu, follow these steps:

1. Select the **Delete** command.
2. Insert the disk that contains your Custom Menu and press **↓** to continue. Follow the directions on the screen (inserting disks if prompted).

- When the Custom Menu is displayed, in the Delete field enter the number corresponding to the file you want to delete, then move the cursor to the Are you sure field? and press **[Y]**.

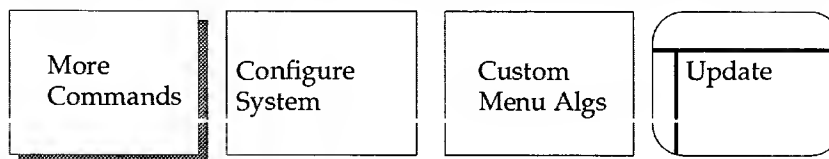
---

**CAUTION:** *If you do not want to delete the file, do not press Enter.*

- To delete the file, press **[J]**. If you do not want to delete the file or When you are done deleting devices, press **[F2]** to return to the Custom Menu Algs menu.

## Update

This command displays the Update screen, which allows you to update the algorithms in a Custom Menu to the current version of the algorithms.



To update algorithms from a Custom Menu, follow these steps:

- Select the Update command.
- Set the following fields using the arrow keys to move from field to field and press **[Space]** to toggle a selected field:

Algorithm media (F,M)	Select <b>F</b> if your current algorithm files are located on Floppy disks or select <b>M</b> if your current algorithm files are located on an MSM (Mass Storage Module). (This field only appears if an MSM is installed.)
Add/Create	Select <b>C</b> (Create) to create a new Custom Menu disk and place the updated version of your old Custom Menu on the new disk.  Select <b>A</b> (Add) to update and add an old Custom Menu to an updated Custom Menu.
New Custom Menu algorithm disk	Select the drive letter that will contain your updated Custom Menu.
Old Custom Menu algorithm disk	Select the drive letter that will contain your old Custom Menu.

---

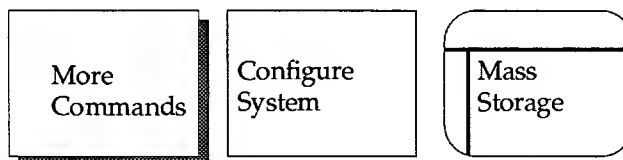
*Note: The algorithm source is always D (default) for an Update operation.*

- Insert the disk that contains your old Custom Menu and press **[J]** to continue. Follow the directions on the screen (inserting disks if prompted).
- To update the Custom Menu, press **[J]** (if you do not want to update the Custom Menu, press **[F2]**).

When you are done updating Custom Menus, press **[F2]** to return to the Custom Menu Algs menu.

## Mass Storage

Use this command to install the system software and algorithm files on the Mass Storage Module (if one is installed in your programmer) and configure your programmer to boot from the MSM.



---

*Note: For information on how to set up an MSM, refer to Chapter 2, "Setup and Installation."*

To install or update system software and algorithm files, do the following:

1. In the Mass Storage screen, use the arrow keys to move from one input field to the next.
2. Set the field to Y (yes) for all the files you wish to install or update on your MSM. You can press the space bar to toggle the selected field.
3. Press ☐ to start the installation or update.
4. Follow any instructions displayed on the screen. For instance, if the message `Insert Algorithm Set 3 Disk` appears, insert the Algorithm disk containing Algorithm Set 3.

---

**CAUTION:** *Do not remove disks from UniSite during this operation unless UniSite prompts you to do so.*

When the mass storage update is done, `OPERATION COMPLETE` is displayed in the message bar.

If you installed the new algorithm files on the Mass Storage Module, the `Update Mass Storage algorithm` prompt should now contain the version number of the new algorithm files in parenthesis.

If you installed the new system files on the Mass Storage Module, the `Update Mass Storage system` prompt should now contain the version number of the new system software in parenthesis.

## Device Checks

The commands on the Device Checks menu allow you to check devices you want to program and to check data in user memory. Commands available on this menu include the following:

- Sumcheck Display
- Compare Electronic ID
- Illegal Bit Check
- Blank Check
- Electronic Erase
- Under/Over-Blow (Logic Devices Only)

Before you can execute a command described in this subsection, you must do the following:

1. Select a device. For more information, see the section titled "Select a Device" earlier in this chapter.
2. Insert and lock a device into the socket. (This applies only if you are checking a device, not if you are checking User RAM.)

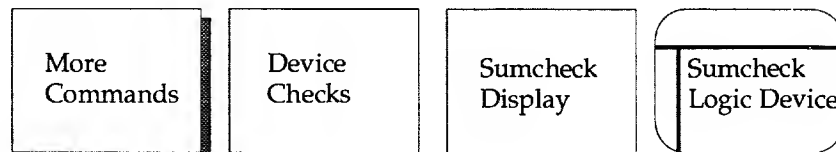
## Sumcheck Display

The sumcheck is a 4- or 8-digit hexadecimal number that, when compared to the original data, allows you to verify that a copy of the data matches the original data. Remember, you must select a device before you calculate the sumcheck. The sumcheck is computed by adding each 8-bit byte in the specified data range into a 32-bit result with the carry dropped.


Below, the options for sumchecking logic and memory are described, with logic devices explained first, followed by memory devices.

### Sumcheck Logic Device

If you have selected a logic device, then the Sumcheck Logic Device screen appears.



To sumcheck a logic device, follow these steps:

1. Select and socket a logic device.
2. Enter the parameters described below.
3. Press  and UniSite calculates the 4-digit sumcheck of the fuse pattern. The sumcheck is displayed in the message bar.

The following parameters appear on this screen:

- **Source (R,D)** — Selects the source of the data to be sumchecked. Press **Space** to toggle between R (RAM) and D (disk).

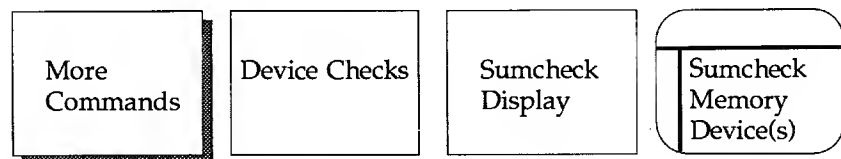
---

**CAUTION:** *Reading the sumcheck file from disk uses RAM as a temporary storage buffer and alters the contents of RAM.*

- **Filename** — Specifies the name of the disk file to sumcheck. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **a:16r8.dat**.

### Sumcheck Memory Device

For memory devices, UniSite calculates and displays the sumcheck for a single device, for each device in a set, or for an entire set.



Follow the steps below to calculate the sumcheck for a memory device:

1. Select and socket a memory device.
2. Enter the parameters described below.
3. Press **↵** and UniSite calculates the 8-digit sumcheck. The sumcheck is displayed in the message bar.

The following parameters appear on this screen:

- **Source (R,D)** — Selects the source of the data to be sumchecked. Press **Space** to toggle between R (RAM) and D (disk).

---

**CAUTION:** *Reading the sumcheck file from disk uses RAM as a temporary storage buffer and alters the contents of RAM.*

- **Filename** — Specifies the name of the disk file to sumcheck. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **27C256.dat**.
- **Sumcheck Entire RAM** — In addition to calculating the sumchecks for the device and set, this option specifies that the sumcheck operation calculates the sumcheck of the entire RAM.
- **Memory Begin Address** — Specifies the first address, in hex, of the first byte of data to be sumchecked. If the source is RAM, the Memory Begin Address is a beginning RAM address; if the source is disk, the Memory Begin Address is a beginning disk file address. The default address is 0.

- **User Data Size** — Specifies the hexadecimal size of the data block to sumcheck. This value is normally equal to the device size or a multiple of device size for sumchecking a set. Entering 0 resets User Data Size to the device size for sumchecking RAM or the file size for a sumchecking a data file.
- **Data Word Width** — Sets the word width, in bits, of the data to be sumchecked. For 8-bit (or larger) devices, the minimum word width is equal to the device word width and the maximum is 64. For 4-bit devices, valid choices are 4, 8, 16, and 32. Data Word Width should match the word width of the data bus in the target system for the device being programmed.
- **Total Set Size** — Specifies how many virtual devices are in the set to be sumchecked. Either enter a number between 1 and 99 or change one of the following parameters and UniSite will calculate the Total Set Size: Memory Begin Address, User Data Size, or Data Word Width. UniSite uses the following equation to calculate the Total Set Size.

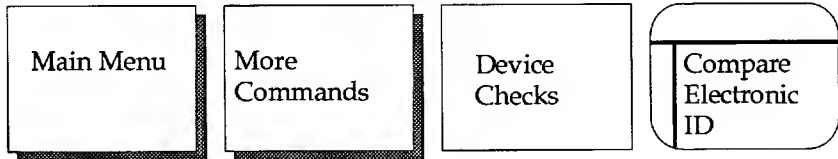
$$\text{TOTAL SET SIZE} = \frac{\text{USER DATA SIZE}}{\text{DEVICE BLOCK SIZE} \left( \frac{\text{DATA WORD WIDTH}}{\text{DEVICE WIDTH}} \right)}$$
0987-1

- **Next Operation Begins At** — A read-only field that specifies the address where the next sumcheck will start. This value is calculated from the Data Word Width, Device Block Size, Memory Begin Address, and device width.
- **For Member X of Y** — X specifies which device in the set is being sumchecked. Y, which is a read-only field, indicates how many devices are in the set(s). Values for X range from 1 to Y.
- **Individual Sumcheck** — A read-only field that displays the individual sumcheck for device number X.
- **Set Sumcheck** — A read-only field that displays the sumcheck for the entire set of Y devices.

## Compare Electronic ID

This feature compares the electronic ID of a device with the electronic ID specified in the selected algorithm, helping prevent accidental damage to a device.

*Note: This command is not supported for devices selected by family/pinout code.*




**CAUTION:** *Turning this feature off could enable you to program a device with the wrong algorithm and could result in damage to that device. We recommend that you keep this feature turned on at all times.*

To compare the electronic ID of a device with the ID stored in the selected algorithm, follow these steps:

1. Select and socket a device that supports electronic ID testing.

*Note: You cannot use an electronic ID to automatically select the proper algorithm to program a device. You also cannot use this feature on devices that do not support electronic ID testing.*

2. Press  and UniSite compares the electronic ID of the socketed device against the electronic ID of the selected device. If the electronic ID of the socketed device matches the electronic ID of the selected device, you see the following message in the message bar:

OPERATION COMPLETE. Device = *ssssssss*

where *ssssssss* is the electronic ID of the socketed device.

3. If UniSite detects an electronic ID that does not match the selected device type, you see the following message in the message bar:

OPERATION FAILED: Electronic ID verify error. Device = *ssssssss*

where *ssssssss* is the electronic ID of the socketed device.



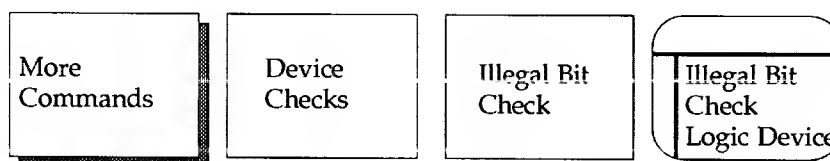
## Illegal Bit Check

The Illegal bit test compares data in a device against data in UniSite's RAM to determine if the device has already-programmed locations of incorrect polarity. For example, UniSite returns an illegal-bit error in the following situation: data in RAM indicates a specific bit should be in an unprogrammed state while the corresponding bit in the device is in a programmed state. The device cannot be programmed if UniSite detects an illegal bit.

If UniSite detects an illegal bit, it displays an error message. If the device is erasable, the illegal bit can be corrected by erasing the device. The Illegal Bit Check is supported for both logic and memory devices, but is not supported for electronically erasable devices.

### Logic Device Illegal Bit Check

If you have selected a logic device, the Illegal Bit Check screen for logic devices appears.



To check a logic device for illegal bits, follow these steps:

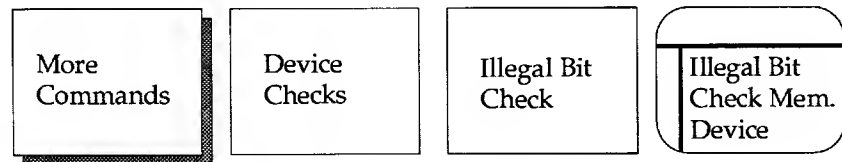
1. Select and socket a logic device.
2. Enter the parameters described below.
3. Press  and UniSite begins the Illegal Bit Check. The results are displayed in the message bar.

The following parameters appear on this screen:

- **Source (R,D)** — Select the source of the data to be checked. Press  to toggle between R (RAM) and D (disk).
- **Filename** — Specifies the name of the disk file to use as the Source. This option appears if you specify disk as the data source. The filename must follow standard DOS conventions, and can include a drive designator. An example of a valid filename is **20V10.dat**.

## Memory Device Illegal Bit Check

If you have selected a memory device, then the Illegal Bit Check screen for memory devices appear.



To check a memory device for illegal bits, follow these steps:

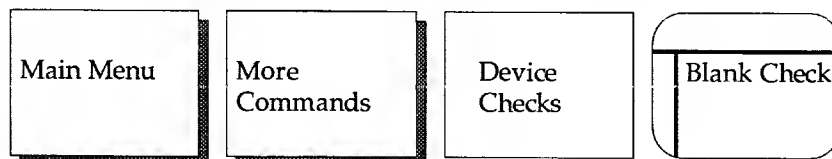
1. Select and socket a memory device.
2. Enter the parameters described below.
3. Press  and UniSite begins the Illegal Bit Check. The results are displayed in the message bar.

The following parameters appear on this screen:

- **Source (R,D)** — Select the source of the data to be checked. Press  to toggle between **R** (RAM) and **D** (Disk).
- **Filename** — Specifies the name of the disk file to check. This option appears if you specify disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **20V10.dat**.
- **User Data Size** — Specifies the size of the data block to check for illegal bits. This value is normally equal to the device size or a multiple of device size for checking illegal bits of a set. If you enter 0, it is reset to device size for RAM or file size for a file operation. User Data Size works with Total Set Size to determine the total amount of bytes to check in a set.
- **Total Set Size** — The total number of parts in the set to check for illegal bits.
- **Data Word Width** — Sets the number of bits in the Data Word Width. For 8-bit (or larger) devices, the minimum word width is equal to the device width, and the maximum is 64. For 4-bit devices, your word width choices are 4, 8, 16, and 32. This value should match the data bus word width in the target system for the device being programmed.
- **Next Device** — Type the number corresponding to the next device in the set to check for illegal bits.
- **Next Operation Begins At** — This field is read-only and cannot be altered. It appears only to inform you where (what hex address) the next operation begins.

## Blank Check

The Blank Check command checks a device, ensuring that it is blank.



To blank check a device, follow these steps:

1. Select and socket a device.
2. Press . UniSite checks the device and respond with

OPERATION FAILED: Non-blank device.

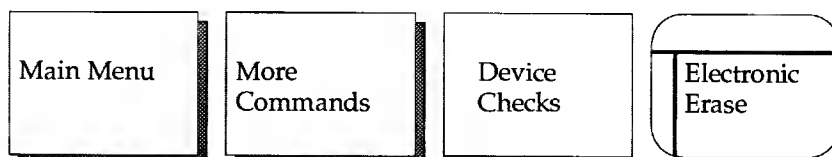
if the device is non-blank, or

OPERATION COMPLETE

if the device is blank.

## Electronic Erase

This command bulk erases an electronically erasable device.



To erase a device, follow these steps:

1. Select and socket an electronically erasable device.
2. Press  to erase the device. When finished, UniSite displays the following message in the message bar:

Done.

3. If you try to erase a device that cannot be electronically erased, UniSite displays the following error.

Electronic bulk erase not supported by device.

---

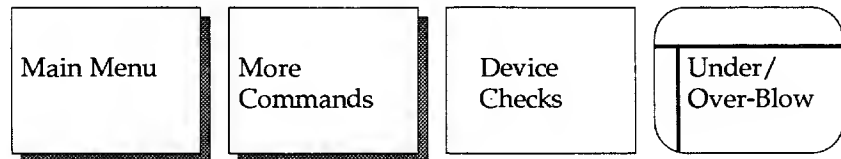
*Note: You cannot access this screen if you have selected a device that cannot be electronically erased (a bipolar PROM, for example).*

*A blank check is run after a bulk erase operation if the blank check switch is enabled and if the selected device supports blank check.*

This device operation is not necessary for most electronically erasable devices. The Electronic Erase is part of the normal programming cycle. Before programming an electronically erasable device, UniSite checks the device and displays a warning if the device is non-blank. If you enable the erasing of the device, UniSite erases the device before programming the device.

**Under/Over-Blow (Logic Devices Only)**

The under/over-blow feature compares the fuse map of a logic device with the fuse map in RAM or in a disk file.



An underblow condition means that the device's fuse is intact, but the data in memory indicates that it should have been blown. An overblow means that the device's fuse is blown, but should have remained intact. (The under/overblow feature is not supported for POF devices.)

To use the under/overblow feature, follow these steps:

1. Select and socket a logic device.
2. Enter the parameters described below.
3. Press **[J]** to begin the test. The under/over-blow screen is displayed. If the data source does not have proper fuse data (for the specified device type), a message appears saying that the file is not initialized. Type **[C]** to initialize the file.
4. With two exceptions, the data shown on the screen is displayed in a format similar to that of the fuse editor. The first exception is that additional character symbols are used to display overblown (B) and underblown (U) data. The second is that unlike the fuse editor, no data can be edited. The fuse number corresponding to the cursor's location appears at the top of the screen. To move the cursor, use the arrow keys. The editor commands are described later in this chapter.

The available parameters are described below.

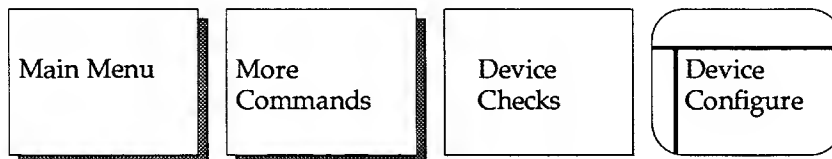
- **Source (R,D)** — Select the source of the data to be compared against that of the device. Press **[Space]** to toggle between **R** (RAM) and **D** (disk).
- **Filename** — Specifies the name of the disk file to compare to the device. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **b:16r8.dat**.

**Under/Overblow  
Commands**

Command	Keystrokes	Description
Next Block	<b>Ctrl</b> + <b>N</b>	Displays the next page of under/overblow data.
Prev Block	<b>Ctrl</b> + <b>P</b>	Displays the previous page of under/overblow data.
Jump to Fuse	<b>Ctrl</b> + <b>B</b>	Moves the cursor to a specific fuse. A highlighted area appears just after the "^B: Jump to Fuse" prompt at the bottom of the screen. Type in the fuse number that you want to jump to and press <b>↵</b> .
Search Pattern	<b>Ctrl</b> + <b>F</b>	<p>Searches for one of four character symbols within the under/overblow data. The four characters you can search for are:</p> <p>X (intact fuse) - (blown fuse) B (overblown fuse) U (underblown fuse)</p> <p>After you select the search character, the search begins at the current cursor position and continues until either a match is found or the end of the fuse map is reached.</p>
Exit	<b>F2</b>	Exits the Under/Overblow screen and returns UniSite to the Device Checks menu.

### Device Configure (Devices That Support Sector Configuration Only)

The device configure feature allows you to set the switches for erasing, programming, and protecting sectors on devices that support sector configuration. These settings are used during the Program and Electronic Erase operations.



To use the device configure feature, follow these steps:

1. Select a device that supports sector configuration.
2. If you wish to enable electronic erase, set the Erase EE device field to YES in the Program screen.
3. On the Device Configure screen, use the arrow keys to move from field to field and press **Space** to toggle between Y (yes) and N (no).

The following fields exist for each sector:

- **Erase** — Set to Y if you wish the sector to be erased when a device erase operation is performed. Set to N to disable erase on the sector.
  - **Program** — Set to Y if you wish the sector to be programmed when a device program operation is performed. Set to N to disable programming on the sector.
  - **Protect** — Set to Y if you wish the sector protect to be enabled. Note that not all devices that support sector erase and program also have support for sector protect. Set to N to disable protect on the sector.
4. When you are finished setting the erase, protect, and program fields, press **F2** to return to the Device Checks menu or press **F1** to return to the Main Menu.

## Edit Data

Use the commands on the Edit Data menu to make changes to data stored in RAM or to data stored in a disk file. When you select the Edit Data command, UniSite displays a menu corresponding to the type of device that is currently selected. There are separate editors for memory and logic devices. For logic devices, there is a fuse map and test vector editor. For memory devices, there is a memory editor.

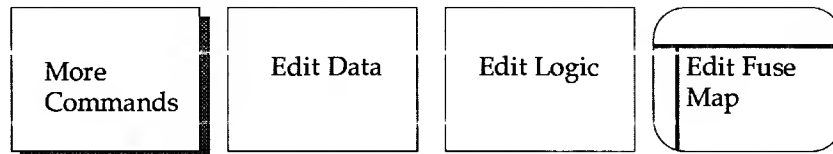
The commands on the Edit Logic menu is explained first, followed by the Edit Memory menu.

## Edit Logic Menu

The Edit Logic menu appears if you have selected a logic device. This menu contains the Edit Logic, Vector Edit, Fill Fuse Map, and Clear Vectors commands.

### Edit Fuse Map

This is the data editor for logic devices.



To edit fuse data, follow these steps:

1. Enter the parameters described below. If necessary, load the data into UniSite.
2. After you have selected the parameters, press **[J]**. The screen clears and show the fuse map data.
3. If the data source does not have proper fuse data, a message appears indicating the file is not initialized. Press **[C]** to initialize the fuse map to an unprogrammed (blank) state.
4. While editing, you can enter either data or commands. To edit the fuse map data, move the cursor to the fuse you want to change. Press **[Space]** to toggle the fuse to the desired state.

The fuse editor commands are described after the parameter list.

---

*Note: In general, any paging command or an exit command causes all currently displayed data to be written to the data source.*

The different options and commands for the editor are explained below.

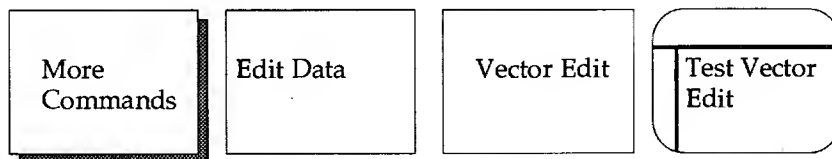
- **Source (R,D)** — Specifies the source of the data to be edited. Press **[Space]** to toggle between **R** (RAM) and **D** (disk).
- **Filename** — Specifies the name of the disk file containing the fuse data to edit. This option appears only if you select disk as the Source. The filename must follow standard DOS conventions.
- **Data Representation (X/-,0/1)** — Specifies how the data in RAM or data file appears on the terminal's screen. The two choices for this parameter are **X** and **-**, or **0** and **1**. Press **[Space]** to toggle between the two options. **X** and **0** represent an unprogrammed state; **-** and **1** represent a programmed state.

The commands described below are available when using the fuse editor.

Command	Keystrokes	Description
Prev Block	<b>Ctrl</b> + <b>P</b>	Displays the previous block of fuse data.
Next Block	<b>Ctrl</b> + <b>N</b>	Displays the next block of fuse data.
Jump to Fuse	<b>Ctrl</b> + <b>B</b>	Moves the cursor to a specific fuse. A highlighted area appears just after the ^B: Jump to Fuse prompt at the bottom of the screen. Type in the fuse number that you want to jump to and press <b>Enter</b> .
Restore Block	<b>Ctrl</b> + <b>U</b>	Returns the current page of fuse data to its original state (before editing that page). Only the data visible on the screen is affected by this command. This command works only if you have not moved off the currently displayed page of edit data since any changes were made.
Exit Editor	<b>F2</b>	Exits the fuse editor.

## Vector Edit

The vector editor allows you to edit test vectors you have created for a particular logic device.



To edit test vectors, follow these steps:

1. Set the parameters for test vector editing. The parameters are described below.
2. Press **Enter**. The screen displays the test vectors (if any) for the selected device.
3. If the source data does not match the device type selected, a message appears indicating the file is not initialized. Type **C** to initialize it.
4. While editing, you can enter either data or commands. You may type only certain test conditions and use only certain keyboard commands in the vector editor. The available editor commands are described after the parameter list.

The available parameters are described below.

- **Source (R,D)** — Specifies the source of the test vectors to be edited. Press **Space** to toggle between R (RAM) and D (Disk).



- **Filename** — Specifies the name of the disk file containing the test vector data to edit. This option appears only if you select disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is `a:16r8.dat`.
- **Edit Begin Vector** — Specifies the first test vector you want to edit. Move the cursor to the Edit Begin Vector field and enter the desired vector number. The vector number you type must be less than or equal to the last vector in RAM or the disk file. This field defaults to 1.

## Test Conditions

The test conditions and the allowed commands are listed in the following tables.

Vector Symbol	Description
0	Drives the specified input pin low.
1	Drives the specified input pin high.
2-9	Super-voltages, defined by the device's manufacturer.
B	Buried register preload.
C	Drives the specified input with a sequence of logic states: in this case, low, high and low (high clock).
D	A single transition that drives the specified input low using a fast slew rate
F	Specifies that a particular input or output pin is to be floated (tri-stated).
H	Verifies that the specified output pin is high.
K	Drives the specified input with a sequence of logic states: in this case, high, low, and high (low clock).
L	Verifies that the specified output pin is low.
N	Specifies that a particular input or output pin is floating (tri-stated). UniSite's F and N conditions perform the same function.
P	Identifies a preload vector and invokes a preload algorithm. This character is allowed on the clock pin ONLY; otherwise, it is treated as an X.
U	A single transition that drives the specified input high using fast slew rate; equivalent to C without returning to the low state. If more than 16 Ds or Us are used in any one test vector, the extra Ds or Us are ignored during test.
X	Ignores the state of an output pin. UniSite applies a logic level specified by a JEDEC file. X field value (1 or 0) or a low is used as the default value.
Z	Verifies the specified input or output pin has high impedance. UniSite will toggle pin using a small current during this test.

---

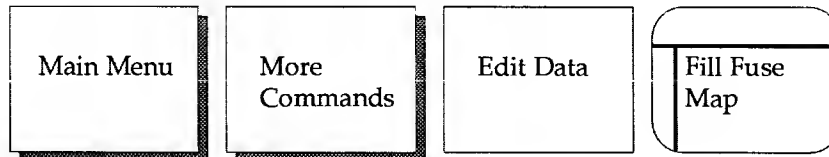
*Note: C, K, U, and D are clock functions that allow setup time.*

## Vector Editor Commands

Command	Keystrokes	Description
Jump to Vector	<b>Ctrl</b> + <b>B</b>	Moves the cursor to a specific vector. A highlighted field appears just after the ^B: Jump to Vector prompt at the bottom of the screen. Enter the vector number to jump to and press <b>J</b> .
Delete Vector	<b>Ctrl</b> + <b>D</b>	Deletes the current vector where the cursor is located.
Insert Default	<b>Ctrl</b> + <b>I</b>	Inserts a default vector, which consists of a vector of all Xs (the character that expresses the ignore input and output test condition). Use the default vector for creating new test vectors. To create a new test vector, insert a default vector and change that vector to contain the test conditions that you specify; the legal test conditions are listed in the previous table. When you enter this command, the default vector is placed in front of the vector highlighted by the cursor.
Next Block	<b>Ctrl</b> + <b>N</b>	Displays the next block of vectors.
Prev Block	<b>Ctrl</b> + <b>P</b>	Displays the previous block of vectors.
Restore Block	<b>Ctrl</b> + <b>U</b>	Restores the current page of vector data to its original state (before editing this page began). Only the data visible on the screen is affected by this command. This command is effective only if there have been no paging commands since changes were made.
Save Vector	<b>Ctrl</b> + <b>W</b>	Saves the current vector (at the cursor) to a temporary buffer.
Repeat Saved	<b>Ctrl</b> + <b>V</b>	Inserts the vector that was last saved using the <b>Ctrl</b> + <b>W</b> command. When you execute this command, the saved vector is placed in front of the vector highlighted by the cursor.
Exit Editor	<b>F2</b>	Exits the vector editor and returns to the previous screen.

## Fill Fuse Map

This command enables you to fill the fuse map with a variable.



This command is useful if you have loaded a fuse map into memory and you want to clear the fuse map from memory.

You can also perform this operation automatically as part of a download operation in two different ways. First, you can use the **F** field in a JEDEC file. Or second, you can enable the Fill Memory option on the Communication Parameters screen.

To fill the fuse map with a variable, follow these steps:

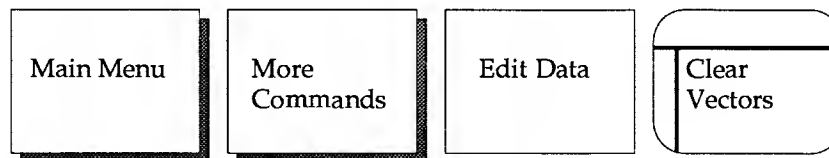
1. Enter the desired one-digit value (0 or 1) in the Fill Variable field. Press **[Space]** to toggle the variable between 1 and 0. 0 represents an unprogrammed state, while 1 represents a programmed state.
2. When the desired fill variable is displayed, press **[Enter]**.
3. UniSite fills the fuse map with the specified variable. When done, UniSite displays

Done

in the message bar.

## Clear Vectors

This command enables you to clear the current test vectors from memory.



To clear vectors, press **[Enter]**.

---

*Note: Only vectors in UniSite's RAM is cleared. This command cannot be used to clear vectors stored on a disk.*

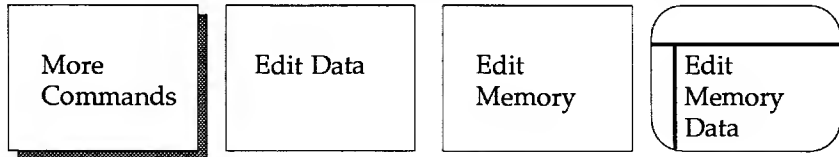
## Edit Memory Menu

The Edit Memory menu appears if you have selected a memory device.

This menu contains the Edit Memory, Complement, Move Data, Fill Memory, and Swap Data commands.

## Edit Memory

Use the Edit Memory command to edit the data for a memory device.



To edit data stored in memory, follow these steps:

1. Specify the memory editing parameters.
2. Press **[Enter]** to enter the editor. Depending on the selected word width, either the 4-, 8-, or 16-bit word width memory editor screen appears.
3. To change data on the screen, move the cursor to the memory location to change and type the new characters over the old ones. Enter the data either in hex or ASCII mode: select the mode using the TAB command (see below). When the cursor is moved around, its location is represented at the top of the screen as a hex address.

The editor commands are described following the parameter list.

The available parameters are described below.

- **Source (R,D)** — Specifies the source of the data to be edited. Press **[Space]** to toggle between **R** (RAM) and **D** (Disk).
- **Filename** — Specifies the name of the disk file containing the data to edit. This option appears only if you select disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **a:27c256.dat**.
- **Edit Data Word Width (4,8,16)** — Selects a 4-, 8-, or 16-bit Data Word Width. Press **[Space]** to toggle between the 4-, 8-, and 16-bit options. If you select **8**, the editor treats all addresses as byte addresses. If you select **16**, the editor treats all addresses as word addresses.
- **Edit Address Offset** — Specifies the address you want assigned to the first byte of data in user memory. Using the address offset can save much calculation time on files written on a host system and then downloaded to UniSite. For example, if your host system data file was written using a begin address of 1000H, you could specify an offset of 1000H. Edit data would then be displayed on UniSite's screen beginning with address 1000H.
- **Edit Begin Address** — Specifies the first address you wish to edit. Enter the 1- to 6-digit hex address. This address must be equal to or greater than the edit address offset. The edit address offset value subtracted from the edit begin address value cannot be greater than the user RAM size.

**Memory Editor  
Commands**

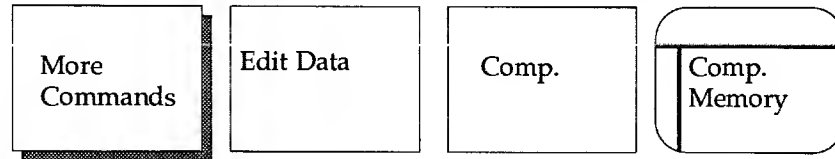
Only certain keyboard commands may be used in the memory editor. The allowed commands are listed in the following table.

Jump to address	<b>Ctrl</b> + <b>B</b>	Moves the cursor to a specific memory address. When this command is selected, the cursor moves to the Jump to Address field. Enter the address you want to jump to and press <b>Enter</b> .
Delete byte	<b>Ctrl</b> + <b>D</b>	Deletes the entire byte with 8-bit data and deletes the entire word with 16-bit data. All the data after the current character position is moved one address position down. The end of RAM has an FF inserted. If a disk file is used, the file gets smaller.
Exchange	<b>Ctrl</b> + <b>E</b>	<p>Allows you to search for a certain pattern and replace it with another pattern:</p> <ol style="list-style-type: none"><li>1. Press <b>Ctrl</b> + <b>E</b>. The cursor moves to the Exchange field at the bottom of the screen.</li><li>2. Type the pattern to search for (any hex value up to 8 digits), followed by <b>Enter</b>. The cursor then moves to the With field at the bottom of the screen.</li><li>3. Type the pattern you want inserted in place of the existing pattern, followed by a <b>Enter</b>. If the pattern cannot be found, a message is displayed and the cursor remains in its original position.</li></ol> <p>When exchange data is entered in 4-bit mode, the upper nibble of data is blank so only up to four characters can be entered in the field.</p>
Next Block	<b>Ctrl</b> + <b>N</b>	Displays the next block of memory data.
Prev Block	<b>Ctrl</b> + <b>P</b>	Displays the previous block of memory data.
Restore Block	<b>Ctrl</b> + <b>U</b>	Returns the current page of data to its original state (before editing began). The page is restored only if there had been no paging commands.

<b>Search Pattern</b>	<b>Ctrl + F</b>	<p>Allows you to search for a particular hex pattern of up to 8 digits:</p> <ol style="list-style-type: none"><li>1. Press <b>Ctrl + F</b>. The cursor moves to the Search field at the bottom of the screen.</li><li>2. Type in the pattern to search for (any hex value up to 8 digits), followed by <b>Enter</b>. If the pattern cannot be found, a message is displayed and the cursor remains in its original position.</li></ol> <p>When searching for data in 4-bit mode, the upper nibble of data is blank so only up to four characters can be entered in the search field.</p>
<b>Start/Stop Insert</b>	<b>Ctrl + T</b>	<p>Toggles the state of the Insert mode between Insert and Overtyping. If the Insert field is displayed in reverse video, the editor is in Insert mode. If the Insert field is displayed in normal video, the editor is in Overtyping mode.</p> <p>In Insert mode, data is inserted at the current cursor position, and all data after that is moved up into higher memory or file addresses. If editing a RAM file, data at the end of RAM is lost. If editing a disk file, the file gets larger. The insert is not complete until the last hexadecimal character (8- or 16-bit) is entered. The cursor moves by byte for 8-bit data or by word for 16-bit data.</p> <p>In Overtyping mode, data entered replaces the current data. When not in insert mode, the arrow keys move the cursor by character.</p> <p>For 8-bit data, the data is entered in bytes, and for 16-bit data, the data is entered in words.</p>
<b>Exit Editor</b>	<b>F2</b>	<p>Exits the memory editor and returns UniSite to the Edit menu.</p>
<b>Toggle Hex/ASCII Modes</b>	<b>Tab</b>	<p>Toggles the mode for data entry. When in hex mode, data is entered on the left side of the screen and the only valid entries are hex characters. When in ASCII mode, data is entered on the right side of the screen and any printable ASCII character can be entered. ASCII mode is not allowed when in 4-bit mode.</p>

## Complement Data

The Complement command converts each bit of data within the specified data block to its opposite value (one's complement).



To complement data stored in memory, follow these steps:

1. Specify the parameters described below.
2. Press  to begin the complement function. UniSite displays the following message when the operation is completed.

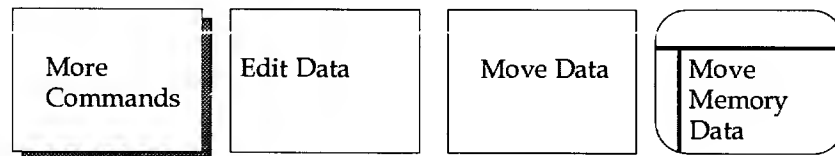
Done

The available parameters are described below.

- **Memory Address** — The memory address at which the complement operation begins. The value entered may be any 1- to 6-digit hex address. The address cannot be greater than the User RAM size.
- **Block Size** — The number of bytes (in hex) that is complemented. Move the cursor to the block size window and enter the block size (from 1 to 6 hex digits). The block size, added to the memory address, cannot exceed the user memory size.

## Move Data

The Move Data command moves a block of data from one location to another.



To move data stored in memory, follow these steps:

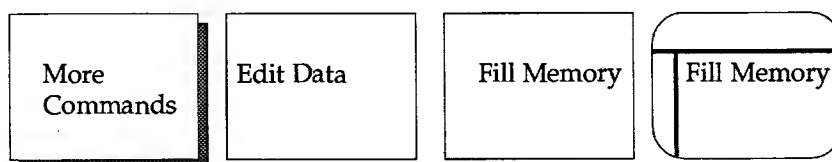
1. Specify the parameters described below.
2. Press  to begin the move function. UniSite displays Done when the move is complete.

The available parameters are described below.


- **From Memory Address** — The first memory address of the data block you want to move data from. Enter any 1- to 6-digit hex address. The address cannot be greater than the user RAM size.
- **To Memory Address** — The first address of the data block you want to move data to. Enter any 1- to 6-digit hex address. The address cannot be greater than the user RAM size.
- **Block Size** — The size (in hex) of the data block to be moved. If the sum of the block size and either the From Memory or the To Memory address values exceed user memory size, a warning message is displayed.

## Fill Memory

The Fill Memory command fills a specified block of data with a 2-digit hex value.



To fill a block of memory, follow these steps:

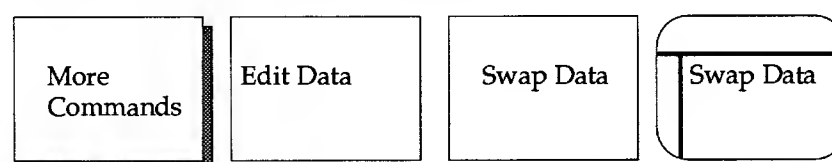
1. Specify the parameters described below.
2. Press  to begin the Fill function. UniSite displays Done when the operation is completed.

The available parameters are described below.


- **Memory Begin Address** — The memory address at which the fill operation begins. Enter any 1- to 6-digit hex address. The address cannot be greater than the user RAM size.
- **Block Size** — The number of bytes (in hex) that is filled. Move the cursor to the block size window and enter the block size (from 1 to 6 hex digits). The block size, added to the memory address, cannot exceed the user memory size.
- **Fill Variable** — The 2-digit hex data variable that is used to fill the specified block. Enter any value between 00 and FF.

## Swap Data


The Swap Data command performs either a byte swap or a nibble swap on the data in a specified block of User RAM.



To swap a block of memory, follow these steps:

1. Specify the parameters described below.
2. Press  to begin the Swap function. UniSite displays Done when the swap is complete.

The available parameters are described below.

- **Swap Mode** — The type of swap to perform. Choose between Byte mode and Nibble mode. In Byte mode, the high byte and the low byte will be swapped. In Nibble mode, the high order nibble will be swapped with the low order nibble. Press  to toggle between the two modes.
- **Memory Begin Address** — The memory address at which to begin the swap. Enter any 1- to 6-digit hex address.



- **Block Size** — The number of bytes (in hex) to be swapped. Enter any 1- to 6-digit hex block size. The block size, added to the memory address, cannot exceed the user memory size. Also, the block size must be an even number if using Byte mode.

## File Operations

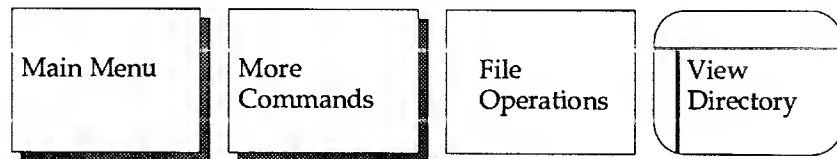
From the File Operations menu, you can access several file manipulation and directory commands. These functions help you move and copy files, view file directories, and organize and maintain your disks. The following sections describe each of the File Operations commands in the order that they appear on the file menu screen.

---

*Note: A file disk has a capacity of 720KB or 112 files, whichever comes first.*

### View Directory

This command displays the file directory of the disk in the disk drive.

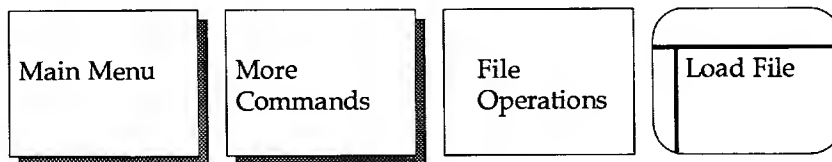


To view a directory, follow these steps:

1. Insert the disk you want to view into the disk drive.
2. Press  to view the directory. The View Directory command can view the directory of any DOS-compatible 3.5" disk. On dual-drive UniSites, the files on drive A are displayed first.
3. UniSite displays 28 files at one time. If your disk(s) contains more than 28 files, they are displayed on the next page(s). Press  +  to advance to the next page of files.

## Load File

This command loads a disk file into RAM. Do not use this command to load files from a PC or from a file server. See the Download Data command for information on transferring files to UniSite from a PC or from a file server.



To load a file into UniSite's RAM, follow these steps:

1. Insert the disk containing the file you want to load into the disk drive.
2. When you select the Load File command, the dialog window displays a directory of the disk in the disk drive. If you do not see your file, press **[F2]** to return to the File Operations menu. Insert the disk containing your file into the disk drive and return to the beginning of this step.
3. UniSite displays 28 files at one time. If your disk has more than 28 files, they are displayed on the next page(s). Press **[Ctrl] + [N]** to advance to the next page of files.
4. Specify the parameters described below. Be sure to include a filename.
5. Press **[Enter]** to begin the loading. Once the disk file is in RAM, you may perform several operations on the file, such as edit or program device; refer to these subsections of the Command chapters for more information on editing and programming.

*Note: The User Data Size field does not appear on the Load File screen. However, this parameter is still updated to reflect the size of the file loaded for use in other screens.*

The available parameters are described below:

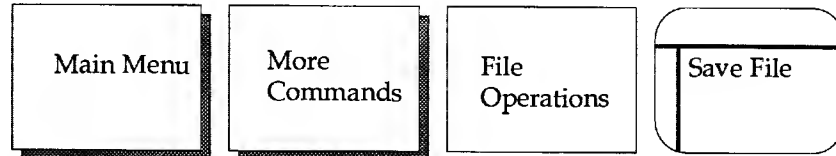
- **Filename** — Specifies the name of the disk file to load. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename would be **b:27c256.dat** or **a:16l8.dat**.
- **Memory Begin Address** — Specifies the first address in memory to load data into. This option appears only if you have selected a memory device. The default address is 0.

If you selected a logic device, the fuse map and vectors from the disk is loaded.

If your files contain data formatted in other than RAM Image Binary (Intel Hex, format 83, for example), use the Transfer Data/Input From Disk command.

## Save File

This command allows you to save the data in RAM to a disk file. Do not use this command to save a file on a PC or a file server. See the Upload Data command for information on transferring files to a PC or a file server from UniSite.



To save data in RAM to disk, follow these steps:

1. Insert the disk you want to save the data to in the disk drive.
2. When you select Save File, the dialog window fills with a directory of the disk in the disk drive. If you do not want to save your file to this disk, press **F2** to return to the File Operations menu. Insert the disk you want to save the data to into the disk drive and return to the beginning of this step.
3. Specify the parameters described below. Be sure to include a filename.
4. Press **↵** to begin the save.

---

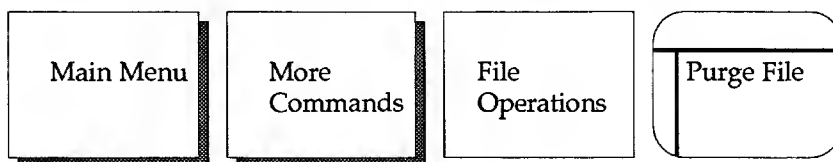
*Note: If you are saving information for a logic device, the fuse map, security fuse state, and vectors are saved.*

- **Filename** — Specifies the name of the disk file to save RAM data into. This may be a new filename, or an existing filename that you want to overwrite. If you are writing to an existing file, the data previously in the file is replaced by the new data. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename would be **a:27c256.dat** or **b:16l8.dat**.
- **Memory Begin Address** — Specifies the first address in RAM to take data to be saved. This option appears only if you have selected a memory device. The default address is 0.
- **User Data Size** — Specifies the size, in hex bytes, of the data block to save. This value is normally equal to the device size. This option appears only if you have selected a memory device.

A saved file is stored in RAM Image Binary format. If you want to store a file in some other format, use the Transfer Data/Output to Disk command.

## Purge File

This command deletes a file, or group of files, from a disk.



To purge a file from a disk, follow these steps:

1. Insert the disk with the file you want to delete into one of the disk drives.
2. When you select the Purge File command, the dialog window fills with a directory listing. UniSite displays up to 28 files at one time. If there are more than 28 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the first page of files.

If you do not see the file you want to delete, press **F2**, insert another disk, and return to the beginning of this step.

3. Move the cursor to the Filename field and enter the name of the file you want to delete. Remember to specify which drive holds the file you want to delete.

---

*Note:* You can use an asterisk (\*) as a wildcard. For example, to purge both 27512.dat and 27128.dat, you could type 27\*.dat.

4. Move the cursor to the Are you sure field and press **Y**.

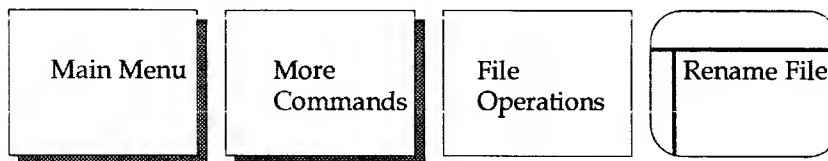
---

**CAUTION:** If you do not want to delete the file, do not press Enter.

5. To delete the file, press **Enter**. If you do not want to delete the file, press **F2** to return to the File Operations menu.

## Rename File

This command changes the name of a file.



To rename a file, follow these steps:

1. Insert the disk with the file you want to rename into one of the disk drives.

---

*Note:* You cannot rename a file to a different disk drive than the current one displayed on the screen.

2. When you select the Rename File command, the dialog window fills with the directory listing. UniSite displays up to 28 files at one time. If there are more than 28 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the first page of files.

If you do not see the file you want to rename, press **F2**, insert another disk, and return to the beginning of this step.

3. Move the cursor to the From field and enter the current name of the file you want to rename. Remember to specify which drive holds the file you want to rename.
4. Move the cursor to the To field and enter the new name for the file you want to rename. Remember to specify which drive holds the file you want to delete.

---

**CAUTION:** If you do not want to rename the file, do not press **Enter**.

5. To rename the file, press **Enter**. If you do not want to rename the file, press **F2** to return to the File Operations menu.

## Copy File

Use the Copy command to copy a file or a group of files.



To copy a file (or group of files), follow these steps:

1. Insert the disk with the file you want to copy into one of the disk drives.

The Copy command can be used on either a single- or a dual-drive UniSite.

2. When you select the Copy File command, the dialog window fills with the directory listing. UniSite displays up to 28 files at one time. If there are more than 28 files, press **Ctrl** + **N** to display the next page of files. Press **Ctrl** + **P** to display the first page of files.

If you do not see the file you want to copy, press **F2**, insert another disk, and go back to step 1.

3. Move the cursor to the From field. Enter the name of the source file. Remember to specify which drive holds the file(s) you want to copy.
4. Move the cursor to the To field and enter the name of the destination file. Remember to specify which drive holds the file(s) you want to copy.
5. Move the cursor to the Single Drive File Copy to Different Disk parameter. If you want to copy the file to a different disk and use the same drive, set this parameter to Y. UniSite prompts you to insert the source disk or destination disk at the appropriate times. This operation results in RAM being used as a temporary storage buffer and alters the contents of RAM

---

*Note: When copying with the Single Drive File Copy to Different Disk parameter set to N, you are prompted to swap disks when necessary. Only one file at a time can be copied using this method.*

Otherwise, if you want to copy the file from one drive to the other on a dual-drive UniSite or to the same disk on either a single- or a dual-drive UniSite, set this parameter to N. Make sure the destination file has a different filename than the source file.

---

*Note: When copying with the Single Drive File Copy to Different Disk parameter set to N, you can copy a group of files by using an asterisk (\*) as a wildcard in the name of the source and destination file.*

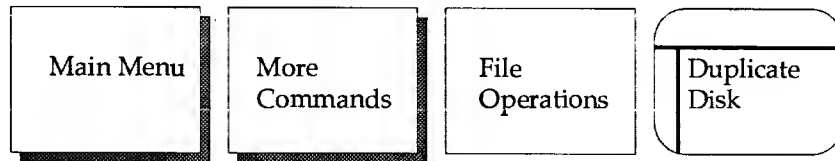
---

**CAUTION:** If you do not want to copy the file, do not press Enter.

6. To begin the copying, press **Enter**. If you do not want to copy the file, press **F2** to return to the File Operations menu.

## Duplicate Disk

Use this command to duplicate an entire disk.



To duplicate an entire disk, you can use either the Duplicate Disk command, or you can use the DOS DISKCOPY command.

## Using DOS

If you have access to a DOS-based PC with a 3.5" disk drive, we suggest that you use the DOS DISKCOPY command to make a copy of your Algorithm disk and System disk. If you use DOS, make sure you use DISKCOPY and not COPY. The backup must be an exact, bit-for-bit, sector-for-sector copy of the original. For more information, see your DOS manual.

---

*Note: The backup disks must first be formatted using the UniSite Format Disk Operation.*

## Using a Dual-drive UniSite

Follow the steps below to duplicate a disk on a UniSite with two disk drives. If your UniSite has one disk drive, skip to the section titled "Using a Single-drive UniSite."

1. Insert the disk you want to duplicate (the source disk) into drive A.

---

**CAUTION:** *Do NOT use an Algorithm disk or the System disk as the destination disk because the original contents of the destination disk will be lost.*

2. Insert the blank disk (the target disk) into drive B.
3. If you want to duplicate the disk, move the cursor to the Are you sure? field and press **Y**.

---

**CAUTION:** *If you do not want to duplicate the disk, do not press Enter; this operation erases the contents of the destination disk. Also, this operation uses RAM as a temporary storage buffer and alters the contents of RAM.*

4. Move the cursor to the Do You Want To Verify Disk field and enable the disk verification procedure.

To disable the verification, and speed up the process, press **N**. To enable the verification, press **Y**.

5. Finally, press **↵** to begin the disk duplication.

---

*Note: After every disk duplication operation, the verification parameter returns to Y and the Are you sure parameter returns to N.*

## Using a Single-drive UniSite

To use a single-drive UniSite to duplicate a disk, follow these steps:

---

*Note: Regardless of whether you use DISKCOPY or Duplicate Disk to duplicate a disk we recommend that you write protect your source disk before you duplicate the disk. To write protect a disk, slide the write protect tab on the disk so you can see through the hole which it exposes. See your DOS user's manual for more information on duplicating a disk.*

1. Insert the disk you want to duplicate into the disk drive.

---

**CAUTION:** *Do NOT use an Algorithm disk or System disk as the destination disk because the original contents of the destination disk will be lost.*

2. If you want to duplicate the disk, move the cursor to the Are You Sure? field and press **Y**.

---

*Note: Because UniSite has only one disk drive, the source disk and destination disk parameters are currently fixed at A.*

---

**CAUTION:** *If you do not want to duplicate the disk, do not press Enter; this operation erases the contents of the destination disk. Also, this operation uses RAM as a temporary storage buffer and alters the contents of RAM.*

3. Move the cursor to the Do you want to verify disk? field and enable the disk verification procedure.

To disable the verification, and speed up the process, press **N**. To enable the verification, press **Y**.

4. Finally, press **↵** to begin the disk duplication.

During the duplication you are prompted when to swap disks.

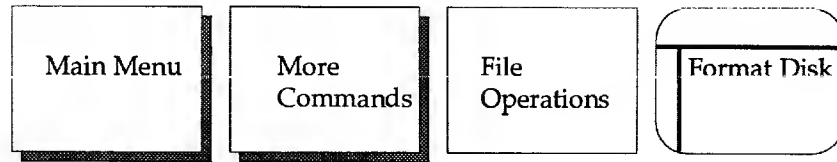
---

*Note: After every disk duplication operation, the verification parameter returns to Y and the Are you sure parameter returns to N.*



## Format Disk

Use this command to prepare a data disk for use. A disk must be formatted before it can be used.



To format a disk, follow these steps:

1. Insert the disk to be formatted into either disk drive.
2. Press **[Space]** to select the drive that the new disk will be in when formatting. For single disk drive systems, you are restricted to drive A.
3. Move the cursor to the **Disk Integrity Check** field and select whether or not you want UniSite to perform a disk integrity check. If enabled, the check is performed while the disk is being formatted, and looks for bad sectors on the disk. Although this test does take more time to complete, it helps ensure the integrity of the newly formatted disk.

To enable the disk integrity check, press **[Y]**. To disable the disk integrity check, press **[N]**.

4. When you are ready to format the disk, type **[Y][J]** at the **Are You Sure?** prompt.

If you are not ready to format a disk, press **[F2]** to return to the File Operations menu.

5. UniSite checks the disk in the disk drive, making sure it is not a System disk. If it is an Algorithm disk or a System disk, UniSite displays the following message:

WARNING: system disk in drive. Hit return to continue,  
^Z to abort.

---

**CAUTION:** *Do NOT format a UniSite Algorithm disk or System disk because the original contents of the disk will be lost.*

To format an Algorithm disk or System disk, press **[J]**. If you do not want to format an Algorithm disk or a System disk, press **[Ctrl] + [Z]**.

## Job File

Using the Job File feature, you can record a series of keystrokes which can then be replayed later.

Job files allow you to perform setup operations without re-keying all the parameters each time a new device is selected. For example, if you regularly program five different devices, you could create and save five different job files, each specifying particular options for a device.

You may store up to 10 job files on each System disk. Each job file may contain up to 499 keystrokes, although a typical job file contains 10 to 20 keystrokes. Job files can be viewed with the View Directory command, where they appear as JFN.JOB, where *N* is a number between 0 and 9. For example, if you have two job files on a disk, they may show up as JF0.JOB and JF2.JOB. If you have job files on another disk, insert the disk and press **[F4]** to display the job file directory for the current disk. Whenever you change disks with job files on them, pressing **[F4]** displays the current job file directory.

### Suggestions on Construction of a Job File

You should not include any Quick Copy commands in your job files. Also, you should not include any operations requiring the insertion or removal of a disk or module. This is because a job file does not stop until the entire file has been played back.

The first command in a job file should be **[F1]**. This way, your job file always starts from the Main Menu, preventing "runaway" job files.

---

*Note: Because screens and key functions may change when the software is updated, a particular job file may only be used with the version of software it was created with. For example, if you created a job file with one version of software, you should re-create the job file so it is compatible with the new software.*

---

### Recording a Job File

To record a job file, follow these steps:

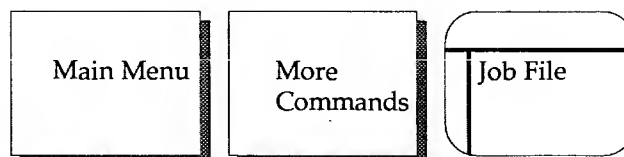
1. Press **[Esc] [Ctrl] + [J]** to start recording the job file. Each keystroke entered from now on is stored in the job file.
2. Press **[F1]** as the first command in your job file. (Although this is not necessary, it helps prevent "runaway" job files.)
3. Enter all the parameters you want recorded. For example, you might want to select a device and then choose its programming parameters by using the Edit Programming Parameters command.
4. When you have entered all the keystrokes you want to store in the job file, press **[Esc] [Ctrl] + [J]** to stop recording the job file. The Job File screen appears.
5. Select a file number to store the newly created job file in. For example, if you want this job file to appear as job file number five on the Job Files screen, press **[5] [Enter]**. If you select a file number already in use, UniSite prompts you to press **[Enter]** to overwrite the existing file. If you want to preserve the existing job file, press **[Ctrl] + [Z]**.

6. Move the cursor down to the **Enter Description** field and type in a name for the job file that has just been recorded. The description can be up to 31 characters long and should be followed by **[J]**.

To save the job file, press **[J]**. While saving, the action symbol rotates. When done, you are returned to the last screen displayed during job file recording.

To not save the job file, press **[F2]** or **[F1]**.

## Playing Back a Job File



Use the Job File command to play back a pre-recorded job file

To play back a job file, follow these steps:

1. Go to the Job Files screen.
2. You see a listing of the job files stored on the disk in the disk drive. To select a job file from this disk, type the number of the job file you want to play back and press **[J]**. For example, to play back the fifth file on the list, press **[5][J]**.

To view a list of job files stored on another disk, insert the disk and press **[F4]**. Pressing **[F4]** reconstructs the job file directory.

3. UniSite now plays back the keystrokes that were recorded. Each screen displayed while you were recording keystrokes is shown (briefly).
4. After the job file is played back, UniSite displays the following message in the message bar:

Job file playback ended

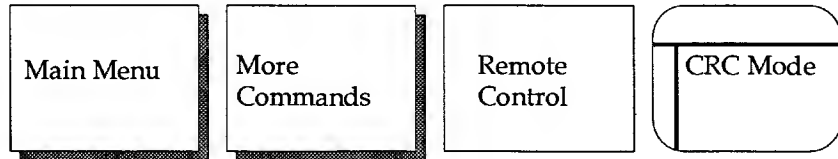
The last screen recorded while you were creating the job file is now displayed. If an error occurred during playback, an error message is displayed and the job file must be re-recorded).

---

*Note: Because screens and key functions may change when the software is updated, a particular job file may only be used with the version of software it was created with. For example, if you created a job file with one version of software, you should re-create the job file so it is compatible with the new software.*

## Remote Control

This command puts UniSite in Computer Remote Control mode.



Chapter 2, "Setup and Installation," provides instructions on how to set up your system. Chapter 6, "Computer Remote Control," provides information on how to use CRC and describes the CRC command set.

To exit remote control, type **Ctrl** + **Z** on the terminal's keyboard, or send a **Z** command from the host computer.

## Self-test

The Self-test command allows you to test circuits and subsystems in UniSite, verifying proper operation or isolating possible problem areas.



An automatic self-test is also performed each time UniSite is powered up. If errors occur during the power-up test, the Self-test screen is displayed, showing the areas that failed.

Some self-tests (such as User RAM test) can be disabled so they are not checked during powerup. Disabling and enabling self-tests is done in the **More Commands...Interface** menu.

---

*Note: For details (help) on the function of each self-test, move the cursor to the test you want information about and press F3.*

### Halting a Self-test

You can stop a self-test anytime during its operation. To halt a self-test, press **Ctrl** + **Z**.

### Running the Self-test

To perform a Self-test, follow these steps:

1. Make sure all device sockets are empty.

---

**CAUTION:** *Executing the System RAM test or the User RAM test erases any data in RAM.*

2. Select the test mode. You can select either one-pass or continuous testing. To toggle modes, move the cursor to the Test Mode prompt and press **Space**. One Pass testing runs the specified test once. Continuous testing runs the specified test until there is a failure or until you halt the procedure by pressing **Ctrl** + **Z**.

---

*Note: There may be a delay before UniSite responds to the Ctrl-Z if the programmer is running the system RAM test.*

3. To test all hardware, move the cursor to the **Perform All Tests** prompt and press . To test a particular item, move the cursor to the desired test and press .

### Interpreting Self-test Results

Four conditions are used as status indicators on the self-test screen:

? UNTESTED

P PASS

F FAIL

- NOT INSTALLED

When testing begins, a ? appears next to the untested areas. As each test completes, either P (pass) or F (fail) appears next to the test name, showing the results of the test performed.

If a hardware item is not installed, a – appears.

For example, if you are testing the Pin Driver boards on a UniSite that has ten Pin Driver boards installed (UniSite can hold 17), ten ? symbols and seven – symbols appear before the testing begins. After testing, UniSite displays ten P symbols and seven – symbols if all the Pin Driver boards pass.

During testing, the message area of the self-test screen indicates that testing is in progress. During the System RAM test, the Remote and the Terminal indicators blink to indicate that testing is in progress.

If ? symbols still appear next to some test names when the testing has completed, it may be because some other test(s) need to pass before the indicated one may be tested. For example, the Waveform board test must pass before the Pin Control Unit test executes. All the installed UniSite hardware must pass self-test.

---

*Note: All of the installed UniSite hardware must pass self-test before any other operations can take place.*

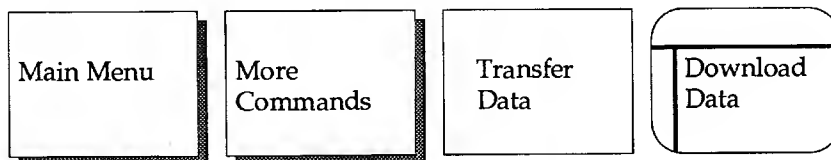
## Transfer Data

Use the commands on the Transfer Data menu to move data files back and forth between UniSite and the host computer.

The Transfer Data menu contains the following seven commands: Download Data, Upload Data, Compare Data, Format Select, Input From Disk, Output to Disk, and Serial Output. Each of these commands is described in this subsection.

### Download Data

Use the Download Data command to specify downloading parameters and to execute the download operation.




Downloading moves a data file from a host computer to UniSite's RAM or disk.

Before you download data, specify the variables for the parameters listed below, then enter a command in the **Download Host Command** field. The information in the command line is a command that your host computer (the computer containing the file to download) recognizes as an instruction to begin the download operation. Finally, press  to execute the download. UniSite displays

Data transfer complete

in the message area when the download is complete.

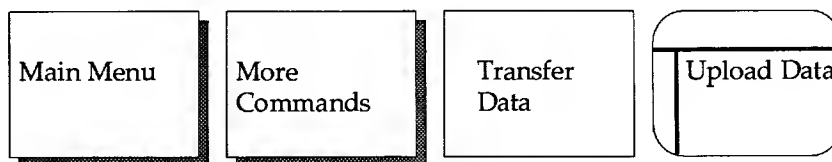
- **Source (R,T)** — Specifies which of UniSite's ports is connected to your host computer. Press  to toggle between R (Remote port) and T (Terminal port).
- **Destination (R,D)** — Specifies the destination of the data that is being downloaded from the host computer. Press  to toggle between R (RAM) and D (disk).
- **Filename** — Specifies the name of the disk file in which to save the downloaded file. This option appears only if you specify disk as the Destination. The filename must follow standard DOS conventions, and can include a drive designator. An example of a valid filename is a:27256.dat.
- **I/O Translation Format** — Selects the translation format of the data in the file. A list of formats UniSite supports is available on the Format Select screen in the Transfer Data menu, and also in the front of Chapter 7 of manual. If you know the number for your format, you can enter it from this screen. If you do not know the correct code number, go to the Format Select screen, find the format you want and enter the correct number from that screen. Entering the format number from the Format Select screen changes the Translation Format parameter on this screen. If you are using the Altera POF format, you must select the desired POF device before you perform a data transfer operation.

- **I/O Addr Offset** — Enter either the beginning hex address of the host computer's data file or the first address you want to capture within a file. This field appears only when a non-JEDEC format has been selected. UniSite subtracts this address from addresses received to determine where, either in the user RAM or in the disk file, the data will be loaded. Entering FFFFFFFF sets the first address received as the I/O offset for the rest of the download.
- **Memory Begin Address** — Specifies the first address, in hex, where the first byte of data is stored from the source port. This field appears only when a non-JEDEC format has been selected. If the destination is RAM, it is a beginning RAM address; if the destination is disk, it is a beginning disk file address. The default address is 0.
- **User Data Size** — Specifies the hexadecimal size, in bytes, of the data block to be downloaded. This field appears only when a non-JEDEC format has been selected. The default is 0, which directs UniSite to receive all the data in the file. After the download is complete, a value equal to the number of bytes received is set here. If a value less than the size of the data received is entered, only the number of bytes equal to that value are actually stored.
- **Download Host Command** — Enter the appropriate host command line here to download the data. This line may be up to 58 characters long. UniSite generates a return character to terminate the line when transmitted to the host. To clear a previously entered command, enter a blank command by pressing **Space** .

If you are using HiTerm, you should use the **TR *filename*** command, where *filename* is the name of the file to download. For more information, see the *HiTerm User Manual* at the back of this binder. For an example of using HiTerm to download data from a PC, see Session 7 in Chapter 4.

## Upload Data

Use the Upload Data command to specify uploading parameters and to execute the upload operation.



Uploading moves a data file from UniSite's RAM or disk to the host computer. To upload a data file, follow these steps:

1. Before you upload data, specify the variables for the parameters listed below.
2. Enter a command in the Upload Host Command field. The information in the command line is a command that your host computer (the computer receiving the data file) recognizes as an instruction to begin the upload operation.
3. Press  to start the upload. During the upload, the action symbol rotates. When done with the upload, UniSite displays the following message in the message bar:

Data Transfer complete. Data sum = xxxxxxxx

The available parameters are described below.

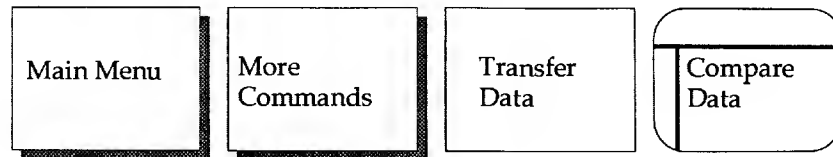
- **Source (R,D)** — Specifies where the data to be uploaded is located. Press  to toggle between R (RAM) and D (disk).
- **Filename** — Specifies the name of the disk file to upload to the host. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can include a drive designator. An example of a valid filename is `b:27256.dat`.
- **Destination (R,T)** — Specifies which port the data file is sent through. Press  to toggle between R (Remote Port) and T (Terminal Port).
- **I/O Translation Format** — Specifies the translation format in which the file is to be generated. The format specified here must be the same as that expected by the host computer. A list of formats supported by UniSite appears on the Format Select screen in the Transfer Data menu, and also in the front of Chapter 7 of manual. If you know the number for your format, you can enter it from this screen. If you do not know the format number, go to the Format Select screen, find the format you want and enter the number from that screen. Entering the format number from the Format Select screen changes the Translation Format parameter on this screen.
- **I/O Addr Offset** — Enter the beginning address of the upload file. This field appears only when a non-JEDEC format has been selected. This value is added to the address of the data in memory (relative to the Memory Begin Address of 0) and output as the I/O address. A value of FFFFFFFF sets the I/O Offset to 0.



- **Memory Begin Address** — Specifies the first address, in hex, from where the first byte of data is retrieved. This field appears only when a non-JEDEC format has been selected. If the source is RAM, it is a beginning RAM address. If the source is Disk, it is a beginning disk file address. The default address is 0.
- **User Data Size** — Specifies the hexadecimal size, in bytes, of the data block to be uploaded. This field appears only when a non-JEDEC format has been selected. Enter the value of the number of bytes to upload. Entering 0 directs UniSite to upload the entire contents of UniSite's RAM. Or, if Disk is specified as the Source, entering 0 directs UniSite to upload the entire disk file.
- **Upload Host Command** — Enter the appropriate host command line here to direct the host to accept the uploaded data. This line may be up to 58 characters long. UniSite generates a return character to terminate the line when transmitted to the host. To clear a previously entered command, enter a blank command by pressing **Space** | **↵**.

## Compare Data

The Compare Data command compares data in user memory with the data file downloaded from the host computer.



This command is useful if you want to verify that you transferred a complete and accurate copy of a data file.

The current I/O format is used to translate the incoming data from the serial port. (JEDEC format cannot be used with this command.) This operation is identical to a downloading operation, except data is compared with, rather than written to, memory.

1. Before you compare data, specify the variables for parameters listed below.
2. Enter a command in the Download host command field. The information in the command line is a command that your host computer recognizes as an instruction to begin the download operation.

3. Press ☐ to start this command. UniSite displays the following message if the two data files are identical.

Data transfer complete

If the data files are NOT identical, UniSite displays the following message:

Data verify error. Data sum = XXXXXXXX

If data in memory does not correspond with data sent from the host, and the terminal is not on the same port as the port receiving the data from the host, UniSite displays the following message:

compare fail at AAAAAA:XX not YY

where AAAAAA is the address, XX is the memory data, and YY is the host's data.

If the terminal is on the same port, then the following message is displayed:

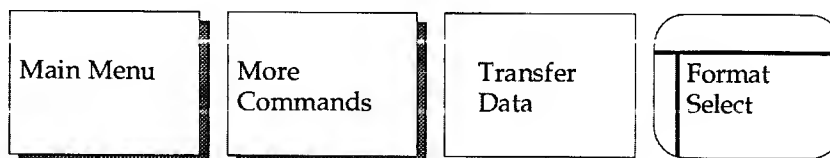
Data verify error. Data sum = ssssssss

- **Source (R,T)** — Specifies which of UniSite's ports is connected to the computer with the data file that is to be used to compare with the memory data. Press ☐ to toggle between R (Remote port) and T (Terminal port).
- **Data Location (R,D)** — Specifies where the data to be compared is located. Press ☐ to toggle between R (RAM) or D (disk).
- **Filename** — Specifies the name of the disk file you want compared. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename would be a:27256.dat.
- **I/O Translation Format** — Specifies the data translation format of the data in the file. A list of formats UniSite supports is available on the Format Select screen in the Transfer Data menu, and also in the front of Chapter 7 of manual. If you know the number for your format, you can enter it from this screen. If you do not know the format number, go to the Format Select screen, find the format you want, and enter the format number from that screen. Entering the format number from the Format Select screen changes the Translation Format parameter on this screen.
- **I/O Addr Offset** — Specifies the beginning address of the downloaded data file to be compared. This field appears only when a non-JEDEC format has been selected. Entering FFFFFFFF causes UniSite to default to the first incoming address as the lowest address to be compared.
- **Memory Begin Address** — Specifies the first hexadecimal address of data to compare with data from the Source port. If the data location is RAM, it is a beginning RAM address. If the data location is Disk, it is a beginning disk file address. This field appears only when a non-JEDEC format has been selected. The default address is 0.

- **User Data Size** — Specifies the hexadecimal size, in bytes, of the data block to be downloaded and compared from the Source to the data location. This field appears only when a non-JEDEC format has been selected. Normally, you should enter a zero here so all of the data is compared. After the compare operation is complete, a value equal to the number of bytes compared is set here. If a value less than the size of the data received is entered, only the number of bytes equal to that value are actually compared.
- **Download Host Command** — Enter the appropriate host command line here to download the data. This line may be up to 58 characters long. UniSite generates a return character to terminate the line when transmitted to the host. To clear a previously entered command, press **Space** then **↵**.

## Format Select

The Format Select command selects the translation format to use.

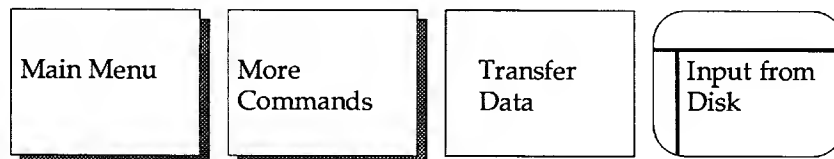


Translation formats (a form of transmission protocol) are used when uploading/downloading data between UniSite and a host computer. Only the formats listed in the Translation Formats chapter of this manual are recognized by UniSite. If your host computer does not generate code into one of the listed formats, edit it to match one of the supported formats.

The Format Select screen displays a list of all UniSite supported formats to choose from. At the bottom of this screen is a format entry field. Enter the number of the format that you want to use and press **↵**. When you select a translation format from this screen, the same format is entered in all of the other Transfer Data screens (i.e., Download Data, Compare Data, etc.).

## Input From Disk

Use the Input From Disk command to load a data file from disk if the data is stored in a translation format. Depending on the settings of the Destination parameter, the data in the disk file is loaded into either RAM or into another disk file.



To input a data file from disk, follow these steps:

1. Insert the disk containing the file to input into the drive.
2. Specify the settings for the parameters listed below.
3. Press  to start this command. UniSite displays the following message when the file has been loaded.

Data transfer complete

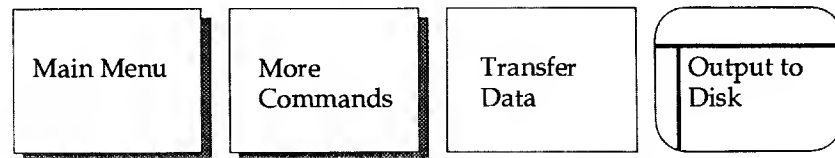
The following parameters appear on the Input from Disk screen:

- **Input Filename** — Specifies the name of the disk file from which formatted data is taken. The filename parameter must follow standard DOS conventions, and can contain a drive designator.
- **Destination (R,D)** — Specifies the destination for the data. Press  to toggle between R (RAM) and D (another disk file).
- **Filename** — Specifies the filename for the disk file into which data is sent. This option appears if you specify disk as the Destination. The filename must follow standard DOS conventions, and can contain a drive designator.
- **I/O Translation Format** — Specifies the format for the data to be input. See Chapter 7 for a complete list of supported formats.
- **I/O Address Offset** — Enter the beginning hex address, or the first address you want to capture within a file, of the disk file's data. This field appears only if a non-JEDEC format has been selected. UniSite subtracts this address from addresses received to determine where the data is loaded into memory. Entering FFFFFFFF sets the I/O Offset equal to the first address received.
- **Memory Begin Address** — Specifies the first address, in hex, to where the first byte of data is stored in memory. If the destination is RAM, it is a beginning RAM address. If the destination is disk, it is a beginning disk file address. The default address is 0. This field appears only if a non-JEDEC format has been selected.

- **User Data Size** — The User Data Size specifies how many bytes (in hex) is read during the Input from Disk operation. The default User Data Size is 0, which causes all of the data to be input. After the operation is complete, UniSite updates the User Data Size parameter to reflect the number of bytes stored to the destination. If a value less than the size of data file input is entered, the number of bytes equal to that value are actually stored. This field appears only if a non-JEDEC format has been selected.

## Output To Disk

The Output To Disk command saves data from a disk file or from RAM to another disk file. The data in the newly created disk file can be stored in any of the supported translation formats.



This command is useful if you already have a data file and want to save it in another translation format. This command is just like the Upload Data command, except that the formatted data is sent to a disk file rather than to a port.

Follow the steps below to output data in a translation format and store it in a disk file.

1. Make sure there is enough space on the disk in the drive to hold the file you are writing.

Use the View Directory command on the File Menu to check the amount of free space on the disk you are going to save the file to.

2. Specify the settings for the parameters listed below.
3. Press ☐ to start this command. UniSite displays

Data transfer complete

when the file has been output to the disk.

The following parameters appear on the Output To Disk File With Translation screen:

- **Source (R,D)** — Select the Source for the disk file data. Press ☐ to toggle between R (RAM) and D (disk).
- **Filename** — Specifies the name of the disk file from which the data is taken. This option appears only if disk is selected as the Source. The filename parameter must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is `b:27256.dat`.
- **Output Filename** — Specifies the name of the disk file you want the formatted data sent to. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is `a:27256.hex`.

- **I/O Translation Format** — Specifies the translation format for the data. A complete listing of the formats is given in the Translation Format chapter of this manual.
- **I/O Address Offset** — Enter the desired beginning address of the disk file. This field appears only if a non-JEDEC format has been selected. UniSite adds this value to the address of the data in memory (relative to the Memory Begin Address of 0) and output the sum as the I/O address. Entering FFFFFFFF sets the I/O Address Offset to 0.
- **Memory Begin Address** — Specifies the first address, in hex, from where the first byte of data is retrieved to write to the disk output file. If the source is RAM, it is a beginning RAM address. If the source is Disk, it is a beginning disk file address. The default address is 0. This field appears only if a non-JEDEC format has been selected.
- **User Data Size** — Specifies the hexadecimal size, in bytes, of the data block to be output, with translation, to the output file from the source. Enter the number of bytes to output. Entering zero sets User Data Size to the total number of hex bytes in UniSite User RAM or the size of the source disk file if disk is used as source. This field appears only if a non-JEDEC format has been selected.

## Serial Output

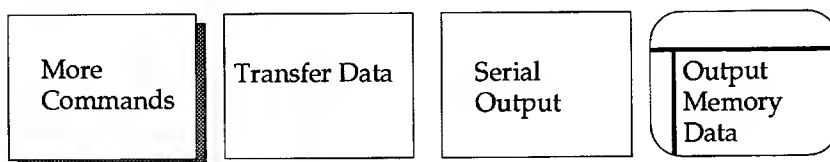
Use the Serial Output command to send data from UniSite to a serial device, such as a printer.

This command is a useful way to obtain a quick copy of programming or other device-related data. Serial Output does not do any data translating. If a logic device is selected, the fuse data is output by fuse number and the vectors is output by vector number. If a memory device is selected, the data is output by address in hex format.

Output Memory Data is explained first, followed by Output Logic Data.

## Output Memory Data to Serial Port

When a memory device has been selected, the Serial Output command is output a specified memory block to one of UniSite's serial ports.



When all parameters described below have been entered, press  to begin the transfer operation.

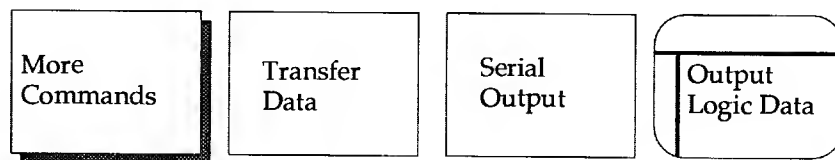
The parameters for this command are listed below.

- **Source (R,D)** — Specifies the source for the data. Press  to toggle between R (RAM) and D (Disk).
- **Filename** — Specifies the disk file to use as the Source. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is `b:27128a.dat`.

- **Destination (R,T)** — Specifies the destination for the data. Press **[Space]** to toggle between R (Remote port) and T (Terminal port).
- **Number Of Lines Between Form Feeds** — Specifies the number of printed text lines you wish to have per page. The default is 0 (no form feed).
- **Memory Begin Address** — Specifies the first address, in hex, of the first byte of data to be retrieved and sent out the serial port. If the source is RAM, it is a beginning RAM address. If the source is disk, it is a beginning disk file address. The default address is 0.
- **User Data Size** — Specifies the hexadecimal size, in bytes, of the data block to be output. Enter the number of bytes to output. Entering zero sets the User Data Size to the total number of bytes in UniSite User RAM or the size of the source disk file if the disk is used as the source.

#### Output Logic Data to Serial Port

If you have selected a logic device, the Output Logic Data to Serial Port screen appears.



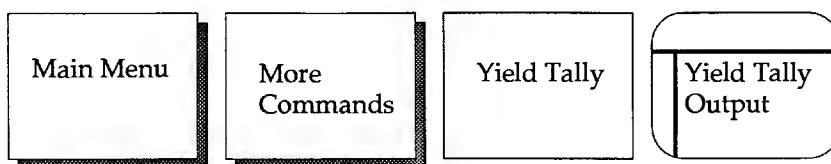
The parameters on that screen are described below:

When all parameters are entered, press **[↵]** to begin the transfer.

- **Source (R,D)** — Specifies the source for the data. Press **[Space]** to toggle between R (RAM) and D (Disk).
- **Filename** — Specifies the disk file to use as the data source. This option appears only if you specify disk as the Source. The filename must follow standard DOS conventions, and can contain a drive designator. An example of a valid filename is **b:1618.dat**.
- **Destination (R,T)** — Specifies the destination for the data. Press **[Space]** to toggle between R (Remote port) and T (Terminal port).
- **Number Of Lines Between Form Feeds** — Specifies the number of printed text lines you wish to have per page. The default is 0 (no form feed).
- **Starting Vector Number** — Specifies the first vector to be output. The default is 1, which causes the vector listing to start at the first vector.
- **Number Of Vectors** — Specifies the total number of vectors you wish to output. The default is 0, which causes no vectors to be output.

## Yield Tally

The Yield Tally command allows you to maintain programming information on devices that have been programmed.



This information can be very useful in a manufacturing environment where device yield statistics must be kept. Yield statistics are maintained on the last 16 device types programmed. If you attempt a yield tally on a 17th device, UniSite drops the statistics for the oldest device. UniSite stores the manufacturer name and its part number or family/pinout codes as the device name in the yield tally record.

The yield data is stored on the System disk in a file called **ytally.sys**. If the file does not exist, enabling the Yield Tally option creates a blank copy of this file on the disk.

---

*Note: You may upload the yield statistics while in CRC mode by using command 43]. The CRC command 46] clears the yield tally statistics. CRC commands are described in Chapter 6, "Computer Remote Control."*

Space is allocated for the Yield Tally data files on the System disk. When the Yield Tally function is invoked without the System disk, the error message

System disk not found. Cannot access yield data.

is displayed on the terminal or, if you are in CRC mode, error code 9A is returned.



**Yield Total**

The Yield total does not include those devices with error conditions that are not recorded in one of the four categories: illegal bit, mis-verify, device not programmable, structured test failure. Examples of these conditions are continuity check, electronic ID error, or overcurrent.

The Yield Tally screen provides statistics for the following categories:

- **Device Name** — Manufacturer's name and part number, family/pinout codes. Statistics for the last sixteen device types are kept.
- **Total Count** — The number of individual devices of the same type that UniSite attempted to program.
- **Good Part** — The number of devices successfully programmed.
- **Illegal Bit** — The number of devices that failed because they did not pass non-blank test or Illegal Bit Check.
- **Verify Fail** — The number of devices that failed because they did not verify.
- **Struct Fail** — The number of logic devices of the same type that failed the logic structured vector test.
- **Device Not Programmable** — The number of devices that could not be programmed because they contained bits that required more programming pulses than were specified.

In the non-blank test, if you stop the operation without programming the device, the illegal bit count is incremented by one. If you proceed with the programming operation, the illegal bit count remains unchanged while the yield tally records the result of the programming operation.

To erase the entire set of statistics, press **Ctrl** + **E** . Press **F2** to go to the previous menu or **F1** to return to the Main Menu. The total number of programming errors is not recorded. This value may be derived by adding the values in the individual error columns.

**Transparent Mode**

Transparent mode can be entered from all of UniSite screens with the following exceptions:

- Editor screens
- Under/over-blow screen
- Yield Tally screen
- Help screens
- CRC mode

With Transparent mode, you can communicate with a host computer connected to one of UniSite's ports. This mode causes the terminal connected to the other port on UniSite to act as if it were connected directly to the host computer. This mode is useful for establishing communication with the host (such as logging in and executing commands).

To enter and exit this mode, type **Esc Ctrl + T** from the terminal. Transparent mode does not support binary data transfers (this can be done via the upload and download commands using one of the binary data formats).

In transparent mode, all key strokes entered on the terminal is passed directly to the host with one exception. The **Esc** character is stripped out since it represents a special meaning to UniSite (it is part of the Exit Transparent Mode command). To send an **Esc** character to the host, enter two consecutive **Esc** characters (the second one is passed to the host), or if **Esc** is followed by some character other than a **Ctrl + T**, the escape and the character is sent to the host.

See Chapter 2 for information on configuring UniSite to operate in Transparent mode.



# 6 *Computer Remote Control*

---

The programmer can be controlled via a host computer using Computer Remote Control (CRC) protocol. CRC commands have been designed to be incorporated into a remote computer software program (driver) which will allow an operator to control the programmer. The driver generates commands and sends them to the programmer, which executes the commands. The programmer then returns a response character, and in some cases, data. The driver reacts to the response and uses it to generate messages and prompts for the user.

---

*Note: You do **not** need to use CRC if you are using PROMlink, TaskLink, or accessing the programmer's built-in menu system (using HiTerm or a similar product to communicate with the programmer). CRC commands offer you an alternative, allowing you to create your own custom interface with the programmer.*

This chapter is not intended to be a complete guide to using CRC commands. For a more detailed explanation of CRC commands, refer to the "UniSystem Computer Remote Control" Application Note available from Customer Support.

This chapter contains the following information:

- **System Setup** — Explains how to set up the programmer for remote control operation. Includes information on entering and exiting CRC mode.
- **CRC Summary** — Gives a listing of the available CRC commands.

## **Which Driver to Use?**

If you are using CRC commands, you must use a driver program to send the CRC commands and receive the programmer's responses. You can either write your own software driver or use an already created driver (such as the `terminal.exe` program included with Windows).

## System Setup

The programmer receives CRC commands and sends responses to the host computer through an RS-232C port using a 25-pin D connector in two possible configurations: either DTE or DCE. Only the Remote port supports CRC operation.

The pin designations for the Remote port are shown in the "More About Cables" section of Chapter 2. Included in that section is a table of pin definitions, which explains the function of each pin for the two serial port configurations.

To ensure correct operation of the Remote port with the host computer, set the parameters for the Remote port according to the host computer requirements.

### Entering CRC Mode

CRC mode can be entered in either of two ways: by the Remote Control menu, or automatically at powerup.

#### By Menu Commands

To enter CRC mode using the Remote Control menu, do the following:

1. Press **F1** to go to the Main Menu.
2. Type **M** to select the More Commands menu.
3. Press **R** to select Computer Remote Control from the More Commands menu.

The programmer is now in Remote Control mode. Except for **Ctrl** + **Z**, all keyboard input will be ignored.

**On Power-up**

The programmer enters either terminal or CRC mode during power-up based on the following combination of port connections and parameter settings.

Parameter Settings		Port Connections		Result
Power Up CRC	User Menu Port	Terminal Connected	Remote Connected	
Off	T (Terminal)	Yes	Yes	Terminal mode on Terminal port
Off	T (Terminal)	No	Yes	CRC mode on Remote port
Off	R (Remote)	X	Yes	Terminal mode on Remote port
On	X	X	Yes	CRC mode on Remote port
X	X	Yes	No	Terminal mode on Terminal port

*Note: X = don't care condition*

If you wish to have the programmer power up in CRC, perform the following steps:

1. Press **[F1]** to get to the Main Menu.
2. Type **[M]** to select the More Commands menu.
3. Press **[C]** to select the Configure System menu.
4. Press **[E]** to select Edit from the Configure Systems menu.
5. Press **[I]** to select Interface from the Edit menu. The programmer displays the interface parameters.
6. Move the cursor to the Power on CRC field and press **[Y]**. CRC is now selected. The following steps in this procedure save CRC mode as a powerup system parameter.
7. Press **[F2]** two times to return to the Configure System Parameters menu.
8. Press **[S]** to select Save from the Configure System Parameters menu. The screen displays the Save System Parameters menu.
9. Type **[1][J]** to select the Powerup Defaults file as the one where system parameters will be saved.
10. Press **[J]** again so that the selection will be saved to the disk. The next time you power up the programmer, it will enter CRC mode automatically.

**Which Interface Mode?**

You can operate the programmer in one of two interface modes: Terminal and CRC. In Terminal mode you use screens and menus to interact with the programmer. In CRC mode you send single-line commands to the programmer and the programmer responds with single line prompts, responses, and error codes.

---

*Note: Terminal mode operations may be run from either the Terminal port or the Remote port. CRC mode operations must be run from the Remote port; CRC will not work on the Terminal port.*

Depending on the equipment you have connected to the programmer, and on the settings of the User Menu Port and Power-on CRC parameters, you can select which mode is available on which port. (The User Menu Port parameter is found on the More Commands/Configure System/Edit/Communication Parameters screen.)

Factory defaults for the programmer are Terminal mode commands sent through the Terminal port and CRC mode commands sent through the Remote port. The factory default for power-up state is Terminal mode.

**Exiting CRC Mode**

Press **Ctrl** + **Z** to exit CRC from an ASCII terminal on the Terminal port.

From a remote computer, send the **Z** command. If you exit remote mode using the **Z** command, the programmer's parameters are set to what they were **before** you entered remote mode. If you exit using **Ctrl** + **Z**, the programmer's parameters are NOT changed.

**Suspending CRC Mode**

CRC Mode can be suspended temporarily to allow you to go into terminal mode to view or change parameter settings or the device data in memory. If the User Menu Port is set to T (Terminal port) press **Ctrl** + **Z** to suspend CRC Mode. If you are controlling the programmer from the Remote port, send the 49] command.

**Halting CRC Operations**

To halt any command or any ongoing CRC operation, use one of the following commands from the Remote port. Neither of the following two commands requires a **]**. Both commands are immediate and both terminate any preceding command operation.

**ASCII**

Command	Hex Code	Description
Esc	1B	Causes the programmer to unconditionally halt any operation except a binary transfer.
BREAK	n/a	Causes the programmer to unconditionally halt any operation in progress. This includes all data communications transfers. The data line must be held in the spacing condition for 110 ms to 700 ms.

## CRC Default Settings

When CRC mode is entered, certain defaults are set prior to accepting any commands. The default settings are outlined below:

Description	Setting
Upload/download port	Remote port
Data source/destination	RAM
Security fuse data (0 or 1)	0
Program security fuse	No
Reject option (commercial or single)	Commercial
Algorithm Source	D (standard algorithms)
Logic verification option	All
Number of verify passes (0,1 or 2)	2
Fill RAM before downloading	No
Illegal bit check option	No
Blank check option	No
Enable yield tally option	No
EE bulk erase option	No
Odd/even byte swap for 16 bit option	No
JEDEC I/O translate DIP/LCC option	Yes
Continuity check option	Yes
Compare electronic signature	Yes
Host command	Blank
I/O address offset	0
I/O format	MOS technology (format 81)
Instrument control code (0,1, 2)	0
I/O timeout	30 seconds
Upload wait	0 seconds
Number of nulls	255
Serial set auto-increment mode	No
Programming mode	single device
Total set size	1
Upload EOF delimiter flag	Disabled
Download EOF delimiter flag	Disabled

If you exit remote mode using the Z command, UniSite's parameters are set to what they were before you entered remote mode. If you exit using **Ctrl** + **Z**, UniSite's parameters are NOT changed.



## CRC Commands

CRC commands are a set of simplified commands for UniSite. The commands are designed to be received from a controlling computer. Because the commands are so simplified, they can be cryptic.

### CRC Command Summary

You send CRC commands to UniSite by typing the command and then pressing the ☐ key. When UniSite receives a CRC command, the command is executed and a response is sent back, followed by a carriage return. If the response is an F, an error occurred. If the response is a ?, UniSite did not understand the command. If the response is a >, the normal CRC prompt, the command executed properly. Some commands respond with both a value and the prompt. For example, UniSite might return 00284295> when you send the Calculate Sumcheck command. In this case, the 00284295 is the sumcheck and the > indicates that the command executed properly. The I, O and C commands perform any data transfer prior to sending the response.

Each command in the CRC command set is summarized in the following tables and then described in more detail on the pages that follow. The command tables are broken up into standard and extended CRC commands. Standard CRC commands are commonly used commands, such as load, program, and verify. Extended CRC commands are more specific device-related commands, such as Set Security Fuse, Fill Fuse Map, and Set Vector Test Options.

---

*Note: While in CRC mode, UniSite recognizes only uppercase characters.*

Except where noted, the commands use the following notation conventions:

- lowercase alphabetic characters indicate arguments that must be specified
- *h* represents a hexadecimal digit.
- *n* represents a decimal digit.
- *xxx...xxx* represents a string of characters.

For example, *nn02*] indicates that you may precede the 02] command with two decimal digits.

Summary of Standard  
CRC Commands

Command	Description	Response
&	Insert Parts Mode	None
-	Invert RAM	>
/	View device error status	XXYY>
hhhhh:	Select device begin address	>
hhhhh;	Select memory block size	> ; (CRC Command)
hhhhh<	Select memory begin address	>
nn=	Select I/O timeout	>
ffppp@ or ffpp@	Select device type	>
cffA	Enter translation format	>
B	Blank check	>
C	Compare to port	>
D	Set odd parity	>
E	Set even parity	>
F	Error status inquiry	HHHHHHHH>
G	Configuration inquiry	RRDD>
H	No operation	>
I	Input from port	>
J	Set 1 stop bit	>
K	Set 2 stop bits	>
nnL	Load RAM from device	>
hhM	Enter record size	>
N	Set no parity	>
O	Output to port	>
nnP	Program device	>
Q	Swap nibbles	>
nnR	Return status of device	AAAAA/BB/C>
nnS	View sumcheck	HHHH>
nnT	Illegal-bit test	>
hhU	Set nulls	>
nnV	Verify device	>
hhhhhhhW	Set I/O offset	>
X	Error code inquiry	HH....HH>
Y	Display parity errors	HHHH>
Z	Exit remote control	none
I	View device family/pinout code	FFPPPP>
\	Move memory block	>
hh^	Clear/fill RAM with data	>

Summary of Extended CRC Commands	Command	Description	Response
	01]	Display system configuration	See Application Note
	<i>nn</i> 02]	Set upload wait time	>
	<i>n</i> 03]	Set device ID verify option	HHHHHHHH> or >
	<i>nn</i> 04]	Set Remote port baud rate	>
	<i>xxx...xxx</i> 05]	Set host command	>
	<i>n</i> 06]	Select data bits	>
	<i>n</i> 07]	Set next set member	>
	<i>n</i> 08]	Select programming mode	>
	<i>nn</i> 22]	Set data word width	>
	<i>n</i> 23]	Select number of verify passes	>
	<i>n</i> 24]	Select security fuse option	>
	<i>n</i> 26]	Specify logic verify options	>
	<i>n</i> 27]	Set/clear enable/disable security fuse	>
	<i>n</i> 28]	Fill fuse map	>
	<i>n</i> 29]	Set reject count option	>
	<i>hh</i> 2A] or <i>hh</i> 2A]	Enable programming options	>
	<i>hh</i> 2B] or <i>hh</i> 2B]	Disable programming options	>
	<i>nh</i> 2C]	Select memory fill option	>
	<i>hh</i> 2D]	Vector test options	>
	<i>nn</i> 2F]	Return 8-character sumcheck	HHHHHHHH>
	<i>xxx...xxx</i> 30]	Set data file name	>
	<i>n</i> 31]	Set data source/destination	>
	<i>xxx...xxx</i> 33]	Select device manufacturer	>
	<i>xxx...xxx</i> 34]	Select device part number for device operations	>
	<i>xxx...xxx</i> 38]	Load file from disk	>
	39]	Delete all RAM files	>
	<i>xxx...xxx</i> 3B]	Delete disk file	>
	<i>n</i> 3C]	Set data transfer port	>
	<i>xxx...xxx</i> 3E]	Select Keep Current algorithm	>
	40] or <i>n</i> 40]	Upload parts list	See Application Note
	<i>n</i> 41]	Upload self-test results	AAA...AA>
	43]	Upload yield tally	See Application Note
	46]	Clear yield tally	>
	49]	Suspend CRC mode	Displays terminal screen
	<i>n</i> 4D]	Select Algorithm source	>
	<i>n</i> 52]	Select media for algorithms (floppy disk or MSM)	>
	54]	Upload device footnote	See Application Note
	55]	Upload device-specific message	See Application Note
	56] or <i>nn</i> 56]	Upload memory verify failure	ddPAAAAAAAAAHhh
	57] or <i>nn</i> 57]	Get checksum of operation	See Application Note
	58]	Upload system ID	HHHH HHHH HHHH>
	5A]	Display list of parameters	See Application Note
	5B]	Clear vector data	>
	5C]	Load system files for CM algorithm disk	>
	5D]	Write system files to CM disk	>
	5E]	Write algorithms to CM disk	>
	<i>n</i> 5F]	Select algorithm source drive for creating CM algorithms	>

Command	Description	Response
60]	Get number of sectors	dd>
n61]	Get sector configuration settings	HHHH HHHH>
nhhh...hhhh62]	Set sector configuration settings	>
xxx...xxx64]	Select device part number for CM (use xxx...xxx33] to select the manufacturer)	>
A7]	Swap bytes	>
DC]	Device check	See Application Note
DF]	View status of sockets	HH HH . . .HH>
EB]	Input JEDEC data from host	>
EC]	Output JEDEC data to host	>
FC]	Restore CRC entry default parameters	>
FD]	Restore user-defined CRC parameters	>
FE]	Save user-defined CRC parameters	>



# 7 *Translation Formats*

---

Translation formats are different ways of encoding the data in a data file. A data file contains the information to be programmed into a device. The data file could contain the fuse pattern and test vectors for a logic device or the data for a memory device.

Generally, the data, such as the fuse pattern for a logic device, is created on a development platform and is then stored in a particular data translation format. When you want to transfer the data file to UniSite, you will need to set up UniSite to handle the right translation format. During download, UniSite translates the formatted data and stores it in user memory as a binary image file.

---

## About this Chapter

This chapter describes all the translation formats UniSite supports. This chapter does not cover how to select an individual translation format. For information on how to select a data translation format, see Session 6 in Chapter 4.

---

*Note: The information in the Session 6 tells you how to select a translation format from the keyboard. For information on how to use Computer Remote Control to select a translation format, see Chapter 6.*

Below you will find a list, in alphabetical order, of all the translation formats supported by UniSite. Following the list is a description and, in most cases, an example of each translation format, presented in order by format number.

Format	Code	Format	Code
ASCII-BNPF	01 (05*)	Fairchild Fairbug	80
ASCII-BHLF	02 (06*)	MOS Technology	81
ASCII-B10F	03 (07*)	Motorola Exorciser	82
Texas Instruments		Intel Intellec 8/MDS	83
SDSMAC (320)	04	Signetics Absolute Object	85
5-level BNPF	08 (09*)	Tektronix Hexadecimal	86
Formatted Binary	10	Motorola Exormax	87
DEC Binary	11	Intel MCS-86 Hex Object	88
Spectrum	12 (13*)	Hewlett-Packard 64000	
POF	14	Absolute	89
Absolute Binary	16	Texas Instruments	
LOF	17	SDSMAC	90
ASCII-Octal Space	30 (35**)	Jedec format (Full)	91
ASCII-Octal Percent	31 (36**)	Jedec format (Kernel)	92
ASCII-Octal		Tektronix Hexadecimal	
Apostrophe	32	Extended	94
ASCII-Octal SMS	37	Motorola 32 bit (S3 record)	95
ASCII-Hex Space	50 (55**)	Hewlett-Packard UNIX	
ASCII-Hex Percent	51 (56**)	Format	96
ASCII-Hex Apostrophe	52	Intel OMF 386	97
ASCII-Hex SMS	57	Intel OMF 286	98
ASCII-Hex Comma	53 (58**)	Intel Hex-32	99
RCA Cosmac	70		

\* This alternate code is used to transfer data without the STX start code and the ETX end code.

\*\* This alternate code is used to transfer data using the SOH start code instead of the usual STX.

---

## Instrument Control Codes

The instrument control code is a 1-digit number that signals or controls data transfers. Specifically, the instrument control code can be used to implement a form of remote control that provides peripherals with flow control beyond that provided by software handshaking. When using computer remote control, the instrument control code is sent immediately preceding the 2-digit format code. The three values of the instrument control code and associated functions are described below.

<b>0 – Handshake Off</b>	<p><b>Input Function:</b> Send X-OFF to stop the incoming transmission. Send X-ON to resume transmission.</p> <p><b>Output Function:</b> Data transmission will be halted upon receipt of an X-OFF character; transmission will resume upon receipt of an X-ON character.</p>
<b>1 – Handshake On</b>	<p><b>Input Function:</b> Transmit an X-ON character when ready to receive data; transmit X-OFF if the receiver buffer is full; transmit an X-ON if the receiver buffer is empty; transmit an X-OFF after all the data is received.</p> <p><b>Output Function:</b> Transmit a PUNCH-ON character prior to data transmission. Data transmission will be halted upon receipt of an X-OFF character and will resume upon receipt of an X-ON character. A PUNCH OFF character is sent when the transmission is completed.</p>
<b>2 – X-ON/X-OFF</b>	<p><b>Input Function:</b> Send X-OFF to stop the incoming transmission. Send X-ON to resume transmission.</p> <p><b>Output Function:</b> Transmit data only after receiving an X-ON character. Data transmission will be halted upon receipt of an X-OFF character; transmission will resume upon receipt of an X-ON character.</p>

---

*Note: X-ON character is a CTRL-Q, or 11 hex.  
X-OFF character is a CTRL-S, or 13 hex.  
PUNCH-ON character is a CTRL-R, or 12 hex.  
PUNCH-OFF character is a CTRL-T, or 14 hex.*



## General Notes

Some information about data translation is listed below:

### Aborting a Data Transfer

To abort a data transfer at any time, type **Ctrl** + **Z** from the terminal. In CRC mode, send **Esc** or **Break**.

### Compatibility

When translating data, you may use any remote source that produces formats compatible with the descriptions listed in this section.

### Formats with Limited Address Fields

Some formats are not defined for use with address fields greater than 64K. Thus, if you transfer a block greater than 64K, the address fields that would be greater than 64K may wrap around and overwrite data transferred in previous data records. Formats 70 through 86, and 90 may exhibit this characteristic.

### Hardware Handshaking

Hardware handshaking may be used if compatible with the host interface by connecting the appropriate lines at the serial port interface.

Hardware handshake (CTS/DTR) is enabled as the default. However, if those signals aren't connected, the programmer senses this and communicates using software handshake (XON/XOFF). UniSite always uses software handshake regardless of whether hardware handshake is enabled.

### Leader/Trailer

During output of all formats except 89 (HP 64000), a 50-character leader precedes the formatted data and a 50-character trailer follows. This leader/trailer consists of null characters. If the null count parameter is set to FF hex, then the leader/trailer is skipped. To set the null count, go to the More Commands/Configure/Edit/Communication Parameters screen and set the Number of Nulls parameter. If in CRC, use the CRC U command to set the null count.

---

*Note: Formats 10, 11, and 89 do not function properly unless you select NO parity, and 8-bit data.*

## ASCII Binary Format, Codes 01, 02 and 03 (or 05, 06 and 07)

In these formats, bytes are recorded in ASCII codes with binary digits represented by Ns and Ps, Ls and Hs, or Os and Is, respectively. See Figure 7-1. The ASCII Binary formats do not have addresses.

Figure 7-1 shows sample data bytes coded in each of the three ASCII Binary formats. Incoming bytes are stored in RAM sequentially starting at the first RAM address. Bytes are sandwiched between B and F characters and are separated by spaces.

**Figure 7-1**  
*An Example of ASCII Binary Format*

```

FORMAT 01 (OR 05) ① BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF ②
BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF
BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF BPPPPPPPPF ③

FORMAT 02 (OR 06) ① BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF ②
BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF
BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF BHHHHHHHHF ③

FORMAT 03 (OR 07) ① B1111111F B1111111F B1111111F B1111111F ②
B1111111F B1111111F B1111111F B1111111F
B1111111F B1111111F B1111111F B1111111F
B1111111F B1111111F B1111111F B1111111F
B1111111F B1111111F B1111111F B1111111F
B1111111F B1111111F B1111111F B1111111F
B1111111F B1111111F B1111111F B1111111F ③

```

### LEGEND

- ① Start Code - nonprintable STX - CTRL B is the optional Start Code
- ② Characters such as spaces, carriage returns and line feeds may appear between bytes
- ③ End Code - nonprintable ETX - CTRL C

0074-2

Data can also be expressed in 4-bit bytes. UniSite generates the 4-bit format on upload if the data word width is 4 bits. Any other characters, such as carriage returns or line feeds, may be inserted between an F and the next B.

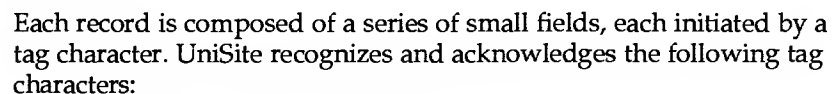
The start code is a nonprintable STX, which is a CTRL-B (the same as a hex 02). The end code is a nonprintable ETX, which is a CTRL-C (the same as a hex 03).

---

*Note: Data without a start or end code may be input to or output from UniSite by use of alternate data translation format codes. These are ASCII-BNPF, 05; ASCII-BHLE, 06; ASCII-B10F, 07.*

A single data byte can be aborted if UniSite receives an E character between B and F characters. Data will continue to be stored in sequential RAM addresses. Data is output in 4-byte lines with a space between bytes.

**Figure 7-2**  
*An Example of TI SDSMAC Format*



- The start-of-file record begins with a tag character and a 12-character file header. The first four characters are the word count of the 16-bit data bytes; the remaining file header characters are the name of the file and may be any ASCII characters (in hex notation). Next come interspersed address fields and data fields (each with tag characters). The address fields represent 16-bit bytes. If any data fields appear before the first address field in the file, the first of those data fields is assigned to address 0000. Address fields may be expressed for any data word, but none are required.

The record ends with a checksum field initiated by the tag character 7 or 8, a 4-character checksum, and the tag character F. The checksum is the two's complement of the sum of the 8-bit ASCII values of the characters, beginning with the first tag character and ending with the checksum tag character (7 or 8).

Data records follow the same format as the start-of-file record but do not contain a file header. The end-of-file record consists of a colon (:) only. The output translator sends a CTRL-S after the colon.

During download or input from disk operations the destination address for the data is calculated in the following manner:

$$\text{Memory address} = (\text{load address} \times 2) - \text{I/O address offset} + \text{begin address}$$

During upload or output to disk operations the load address sent with each data record is calculated in the following manner:

$$\text{Load address} = \text{I/O address offset} / 2$$

The Memory begin address, I/O address offset and User data size parameters represent bytes and must be even values for this format. The upload record size must also be even for this format (default is 16).

---

*Note: If the data will be programmed into a 16 bit device to be used in a TMS320 processor-based system, the odd/even byte swap switch must be enabled.*

---

## The 5-Level BNPF Format, Codes 08 or 09

Except for the start and end codes, the same character set and specifications are used for the ASCII-BNPF and 5-level BNPF formats.

Data for input to UniSite is punched on 5-hole Telex paper tapes to be read by any paper tape reader that has an adjustable tape guide. The reader reads the tape as it would an 8-level tape, recording the 5 holes that are on the tape as 5 bits of data. The 3 most significant bits are recorded as if they were holes on an 8-level tape. Tape generated from a telex machine using this format can be input directly to a serial paper tape reader interfaced to UniSite. UniSite's software converts the resulting 8-bit codes into valid data for entry in RAM.

The start code for the format is a left parenthesis, (Figs K on a telex machine), and the end code is a right parenthesis, (Figs L on a telex machine). The 5-level BNPF format does not have addresses.

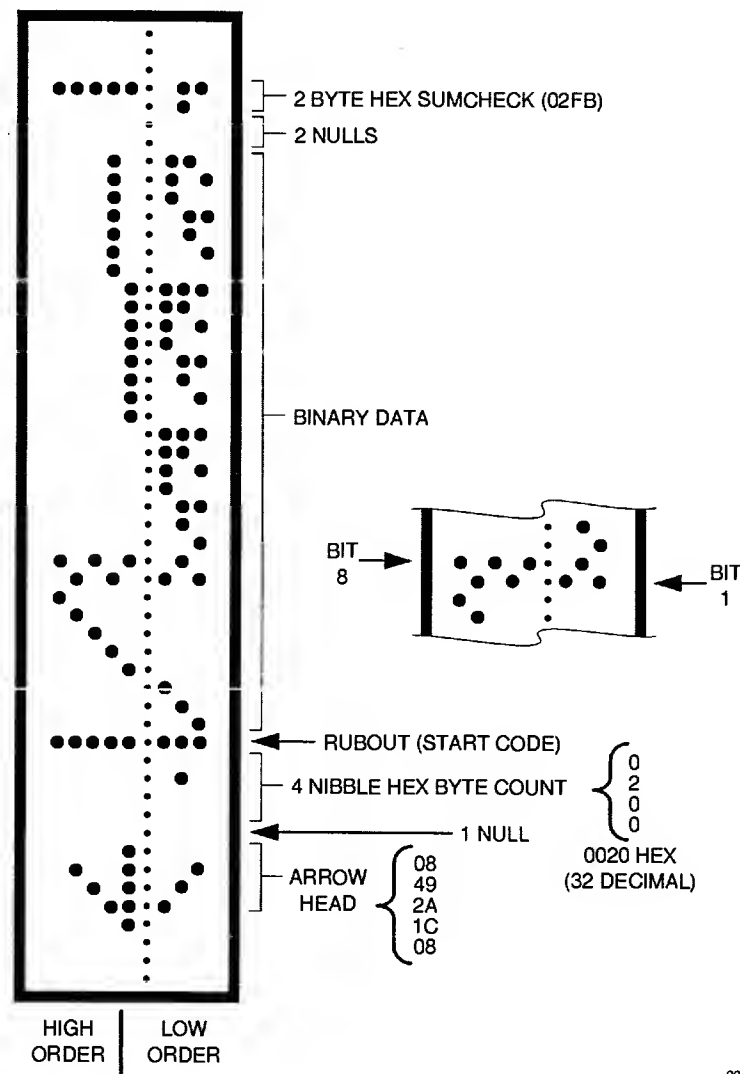
---

*Note: Data without a start or end code may be input to or output from UniSite by use of the alternate data translation format code, 09. This format accepts an abort character (10 hex) to abort the transmission.*

## Formatted Binary format, Code 10

Data transfer in the Formatted Binary format consists of a stream of 8-bit data bytes preceded by a byte count and followed by a sumcheck, as shown in Figure 7-3. The Formatted Binary format does not have addresses.

**Figure 7-3**  
*An Example of Formatted Binary Format*



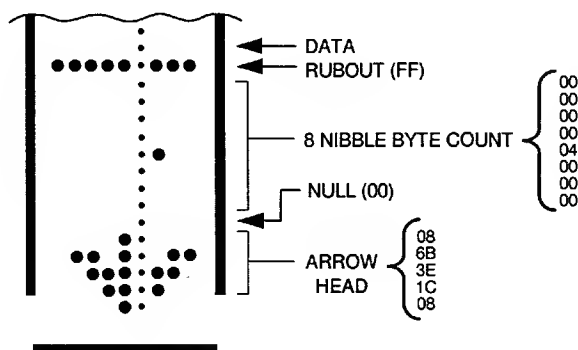
0075-2

UniSite stores incoming binary data upon receipt of the start character. Data is stored in RAM starting at the first RAM address specified by the Memory Begin Address parameter and ending at the last incoming data byte.

A paper tape generated by a programmer contains a 5-byte, arrow-shaped header followed by a null and a 4-nibble byte count. The start code, an 8-bit rubout, follows the byte count. The end of data is signaled by two nulls and a 2-byte sumcheck of the data field. Refer to Figure 7-4.

If the data output has a byte count GREATER than or equal to 64K, an alternate arrow-shaped header is used. This alternate header (shown below) is followed by an 8-nibble byte count, sandwiched between a null and a rubout. The byte count shown here is 40000H (256K decimal). If the byte count is LESS than 64K, the regular arrowhead is used instead. Data that is input using Formatted Binary format will accept either version of this format.

**Figure 7-4**  
*An Example of Formatted Binary Format*



0483-2

In addition, a third variation of this binary format is accepted on download. This variation does not have an arrowhead and is accepted only on input. The rubout begins the format and is immediately followed by the data. There is no byte count or sumcheck.



## DEC Binary Format, Code 11

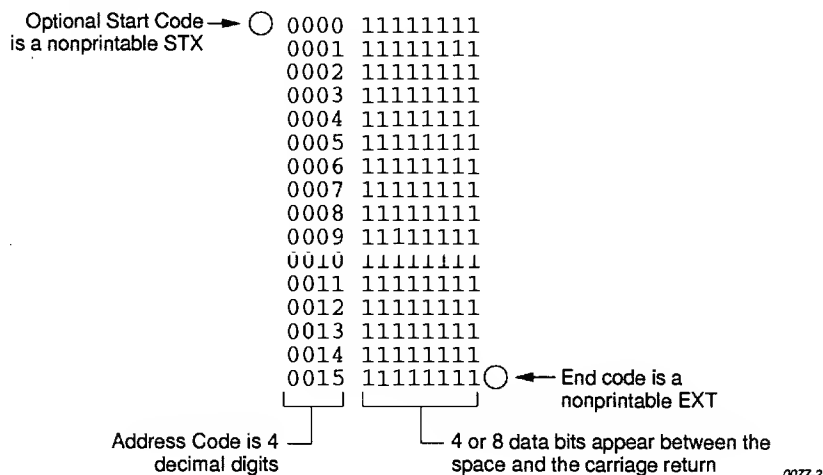
Data transmission in the DEC Binary format is a stream of 8-bit data bytes with no control characters except the start code. The start code is one null preceded by at least one rubout. A tape output from UniSite will contain 32 rubouts in the leader. The DEC Binary format does not have addresses.

## Spectrum Format, Codes 12 or 13

In this format, bytes are recorded in ASCII codes with binary digits represented by 1s and 0s. During output, each byte is preceded by a decimal address.

Figure 7-5 shows sample data bytes coded in the Spectrum format. Bytes are sandwiched between the space and carriage return characters and are normally separated by line feeds. The start code is a nonprintable STX, CTRL-B (or hex 02), and the end code is a nonprintable ETX, CTRL-C (or hex 03).

**Figure 7-5**  
An Example of Spectrum  
Format



*Note: Data without a start or end code may be input to or output from UniSite by use of the alternate data translation format code, 13.*

---

## POF (Programmer Object File) Format, Code 14

The POF (Programmer Object File) format provides a highly compact data format to enable translation of high bit count logic devices efficiently. This format currently applies to MAX™ devices, such as the Altera 5032.

The information contained in the file is grouped into "packets." Each packet contains a "tag," identifying what sort of data the package contains plus the data itself. This system of packeting information allows for future definitions as required.

The POF file is composed of a header and a list of packets. The packets have variable lengths and structures, but the first six bytes of every packet always adhere to the following structure.

```
struct PACKET_HEAD
{
    short tag;           /*tag number - type of packet */
    long length;         /*number of bytes in rest of packet */
}
```

A POF file is read by the program examining each packet and if the tag value is recognized, then the packet is used. If a tag value is not recognized, the packet is ignored.

Any packet except the terminator packet may appear multiple times within a POF file. Packets do not need to occur in numerical tag sequence. The POF reader software is responsible for the interpretation and action taken as a result of any redundant data in the file including the detection of error conditions.

The POF format currently uses the following packet types.

---

*Note: In the following packet type descriptions, one of the terms — Used, Skipped, or Read — will appear after the tag and name.*

*Used: The information in this packet is used by UniSite.*

*Skipped: This information is not used by UniSite.*

*Read: This information is read by UniSite but has no direct application.*

Creator_ID	tag = 1	Used	This packet contains a version ID string from the program which created the POF file.
Device_Name	tag = 2	Used	This packet contains the ASCII name of the target device to be programmed, for example, PM9129.
Comment_Text	tag = 3	Read	This packet contains a text string which may consist of comments related to the POF file. This text may be displayed to the operator when the file is read. The string may include multiple lines of text, separated by appropriate new line characters.

<b>Tag_Reserved</b>	tag = 4	Skipped
<b>Security_Bit</b>	tag = 5	Used This packet declares whether security mode should be enabled on the target device.
<b>Logical_Address_and_Data_16</b>	tag = 6	Read This packet defines a group of logical addresses in the target device and associates logical data with these addresses. The addresses comprise a linear region in the logical address space, bounded on the low end by the starting address and extending upward by the address count specified in the packet.  The starting address and address count are each specified by two-byte fields (16 bits).
<b>Electrical_Address_and_Data</b>	tag = 7	Used This packet defines a group of electrical addresses in the target device and associates data values with those addresses. The data field is ordered in column-row order, beginning with the data for the least column-row address, continuing with increasing row addresses until the first column is filled, then incrementing the column address, etc.
<b>Terminator</b>	tag = 8	Used This packet signals the end of the packet list in the POF file. This packet must be the N-th packet, where N is the packet count declared in the POF header. The CRC field is a 16-bit Cyclic Redundancy Check computed on all bytes in the file up to, but not including, the CRC value itself. If this CRC value is zero, the CRC check should be ignored.
<b>Symbol table</b>	tag = 9	Skipped
<b>Test Vectors</b>	tag = 10	Used This packet allows the POF to contain test vectors for post programming testing purposes. Each vector is a character string and uses the 20 character codes for vector bits defined in JEDEC standard 3A, section 7.0.
<b>Electrical_Address_and_Constant_data</b>	tag = 12	Skipped
<b>Number of programmable elements</b>	tag = 14	Read This packet defines the number of programmable elements in the target device.
<b>Logical_Address_and_Data_32</b>	tag = 17	Read This packet defines a group of logical addresses in the target device and associates logical data with these addresses. The addresses comprise a linear region in the logical address space, bounded on the low end by the starting address and extending upward by the address count specified in the packet.  The starting address and address count are each specified by 4-byte fields (32 bits).

---

## Absolute Binary Format, Code 16

Absolute binary format is a literal representation of the data to be transferred and no translation of the data takes place during the transfer. There are no overhead characters added to the data, i.e. no address record, start code, end code, nulls or checksum. Every byte transferred represents the user's data. This format can be used to download unformatted data (such as a .exe file) to the programmer.

Since this format does not have an end-of-file character, download transfers will terminate after no more data is received and an I/O timeout occurs. This is true for all data formats which don't have an end-of-file character. For this reason do not use a value of 0 for the I/O timeout parameter on the communication parameters screen since this will disable the timeout. A value between 1 and 99 (inclusive) should be used for the I/O timeout parameter when using formats that require the timeout to occur.

---

## LOF Format, Code 17

The Link Object Format (LOF) is an extension of the standard JEDEC data translation format and is used to transfer fuse and test vector data between UniSite and a host computer. LOF is designed to support the Quicklogic QL8x12A family of FPGAs. A LOF data file is stored as an imploded ZIP file, which yields data compression approaching 95%.

---

*Note: The specification for the ZIP data compression algorithm allows for multiple data files to be compressed into one ZIP file. In addition, the ZIP data compression algorithm allows for multiple types of data compression.*

*The UniSite implementation of UNZIP supports only imploded data files and will extract only the first file in a ZIP file. All remaining files in the ZIP file will be ignored, as will all files not stored in the imploded format.*

The LOF format contains both a subset and a superset of the JEDEC format described in this chapter. This section describes only the fields that are extensions of the JEDEC standard or that are unique to the LOF format. See the section explaining the JEDEC format for information on the standard JEDEC fields. See page 7-30 for information on obtaining a copy of the JEDEC Standard 3A.

### LOF Field Syntax

The LOF character set consists of all the characters that are permitted with the JEDEC format: all printable ASCII characters and four control characters. The four allowable control characters are STX, ETX, CR (Return) and LF (line feed). Other control characters, such as Esc or Break, should not be used.

---

*Note: This is Data I/O Corporation's implementation of Quicklogic's Link Object Format. Contact Quicklogic for a more indepth explanation of the format and its syntax.*

**LOF Fields**

The following fields are included in Data I/O's implementation of the LOF format:

<STX>	*	Start of Data (ASCII Ctrl-B, 0x02 hex)
C	*	Fuse Checksum
K		Fuse data, followed by control words and pulse link cycles
N	*	Notes Field
QB		Number of bits per word
QC		Number of control words at the end of each K field
QF		Number of Fuses in Device (# of K fields)
QM		Number of macro cells in the data file
QP	*	Number of Device Package Pins
QS		Number of Hex-ASCII words in each K field and each control word
QV	*	Maximum Number of Test Vectors
R		Signature Analysis (reserved for future use)
S		Signature Analysis (reserved for future use)
T		Signature Analysis (reserved for future use)
V	*	Test Vectors (reserved for future use)
X	*	Default Test Conditions (reserved for future use)
<ETX>	*	End of Data (ASCII Ctrl-C, 0x03 hex)

\* *These fields are already defined as part of the JEDEC standard and will not be defined in this section.*



Although each data byte has an address, most are implied. Data bytes are addressed sequentially unless an explicit address is included in the data stream. This address is preceded by a \$ and an A, must contain 2 to 8 hex or 3 to 11 octal characters, and must be followed by a comma, except for the ASCII-Hex (Comma) format, which uses a period.

UniSite skips to the new address to store the next data byte; succeeding bytes are again stored sequentially.

Each format has an end code, which terminates input operations. However, if a new start code follows within 16 characters of an end code, input will continue uninterrupted. If no characters come within 2 seconds, input operation is terminated.

After receiving the final end code following an input operation, UniSite calculates a sumcheck of all incoming data. Optionally, a sumcheck can also be entered in the input data stream. UniSite compares this sumcheck with its own calculated sumcheck. If they match, UniSite will display the sumcheck; if not, a sumcheck error will be displayed.

---

*Note: The sumcheck field consists of either 2-4 hex or 3-6 octal characters, sandwiched between the \$ and comma characters. The sumcheck immediately follows an end code. The sumcheck is optional in the input mode but is always included in the output mode. The most significant digit of the sumcheck may be 0 or 1 when expressing 16 bits as 6 octal characters.*

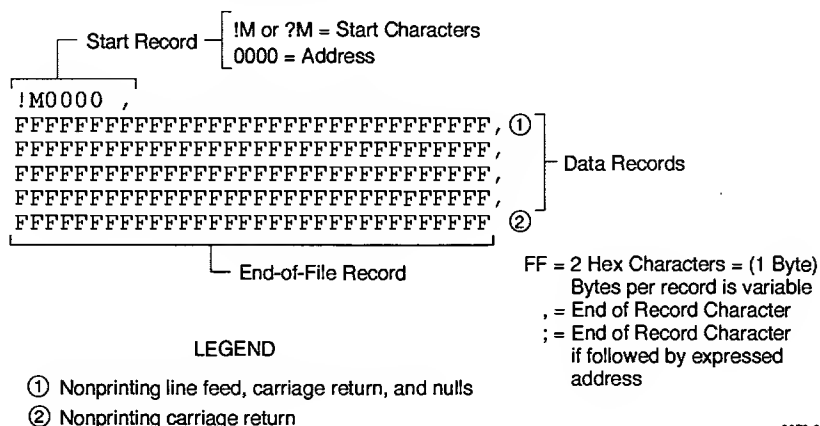
UniSite divides the output data into 8-line blocks. Data transmission is begun with the start code, a nonprintable STX character, or optionally, SOH.\* Data blocks follow, each one prefaced by an address for the first data byte in the block. The end of transmission is signaled by the end code, a nonprintable ETX character. Directly following the end code is a sumcheck of the transferred data.

\* ASCII-Octal SMS and ASCII-Hex SMS use SOM (CTRL-R) as a start code and EOM (CTRL-T) as an end code.

## RCA Cosmac Format, Code 70

Data in this format begins with a start record consisting of the start character (!M or ?M), an address field, and a space. See Figure 7-7.

**Figure 7-7**  
*An Example of RCA Cosmac Format*



0079-2

The start character ?M is sent to UniSite by a development system, followed by the starting address, and a data stream which conforms to the data input format described in the ASCII-Hex and Octal figure. Transmission stops when the specified number of bytes has been transmitted.

Address specification is required for only the first data byte in the transfer. An address must have 1 to 4 hex characters and must be followed by a space. UniSite records the next hexadecimal character after the space as the start of the first data byte. (A carriage return must follow the space if the start code ?M is used.) Succeeding bytes are recorded sequentially.

Each data record is followed by a comma if the next record is not preceded by an address, or by a semicolon if it starts with an address. Records consist of data bytes expressed as 2 hexadecimal characters and followed by either a comma or semicolon, and a carriage return. Any characters received between a comma or semicolon and a carriage return will be ignored by UniSite.

The carriage return character is significant to this format because it can signal either the continuation or the end of data flow; if the carriage return is preceded by a comma or semicolon, more data must follow; the absence of a comma or semicolon before the carriage return indicates the end of transmission.

Output data records are followed by either a comma or a semicolon and a carriage return. The start-of-file records are expressed exactly as for input.



## Fairchild Fairbug, Code 80

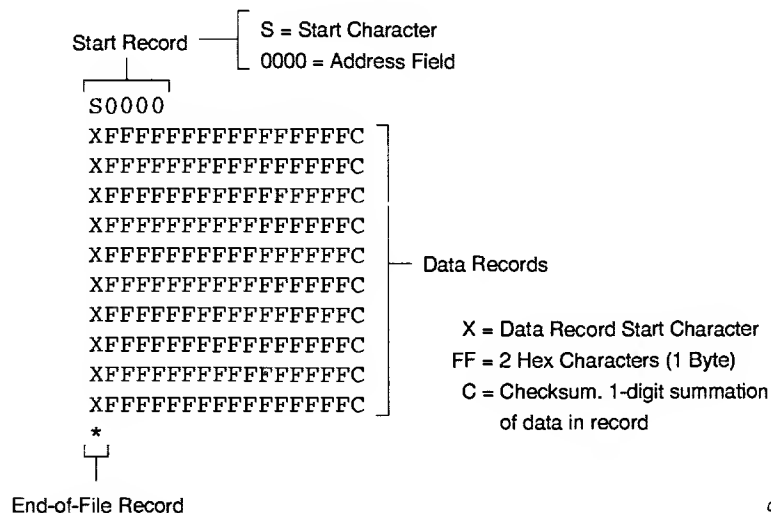
In the Fairbug format, input and output requirements are identical; both have 8-byte records and identical control characters. Figure 7-8 shows a Fairbug data file. A file begins with a 5-character prefix and ends with a 1-character suffix. The start-of-file character is an S, followed by the address of the first data byte. Each data byte is represented by 2 hexadecimal characters. UniSite will ignore all characters received prior to the first S.

*Note: Address specification is optional in this format; a record with no address directly follows the previous record.*

Each data record begins with an X, and always contains 8 data bytes. A 1-digit hexadecimal checksum follows the data in each data record. The checksum represents, in hexadecimal notation, the sum of the binary equivalents of the 16 digits in the record; the half carry from the fourth bit is ignored.

UniSite ignores any character (except for address characters and the asterisk character, which terminates the data transfer) between a checksum and the start character of the next data record. This space can be used for comments.

**Figure 7-8**  
An Example of Fairchild Fairbug



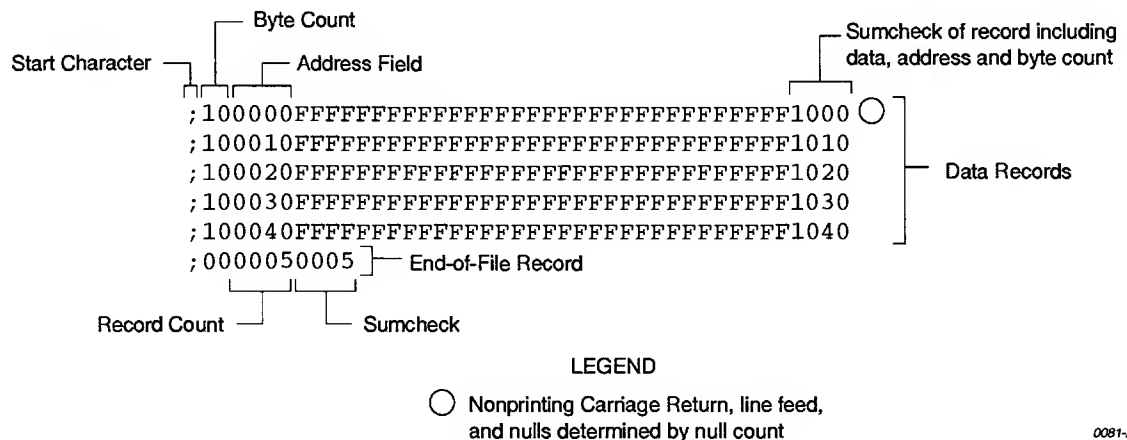
The last record consists of an asterisk only, which indicates the end of file.

## MOS Technology Format, Code 81

The data in each record is sandwiched between a 7-character prefix and a 4-character suffix. The number of data bytes in each record must be indicated by the byte count in the prefix. The input file can be divided into records of various lengths.

Figure 7-9 shows a series of valid data records. Each data record begins with a semicolon. UniSite will ignore all characters received prior to the first semicolon. All other characters in a valid record must be valid hexadecimal digits (0-9 and A-F). A 2-digit byte count follows the start character. The byte count, expressed in hexadecimal digits, must equal the number of data bytes in the record. The byte count is greater than zero in the data records, and equals zero (00) in the end-of-file record. The next 4 digits make up the address of the first data byte in the record. Data bytes follow, each represented by 2 hexadecimal digits. The end-of-file record consists of the semicolon start character, followed by a 00 byte count, the record count and a checksum.

**Figure 7-9**  
*An Example of MOS Technology Format*

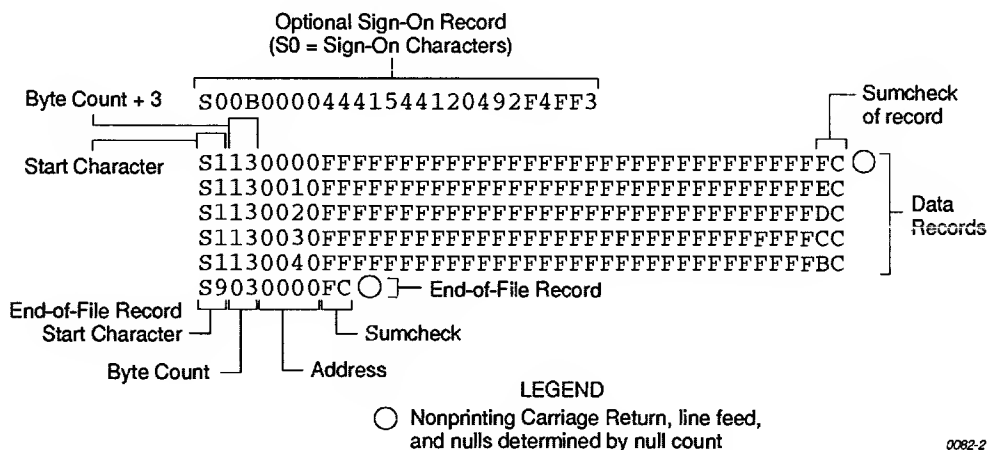


The checksum, which follows each data record, is a 2-byte binary summation of the preceding bytes in the record (including the address and byte count), in hexadecimal notation.

## Motorola EXORciser Format, Code 82

Motorola EXORciser data files may begin with an optional sign-on record, which is initiated by the start characters S0. Valid data records start with an 8-character prefix and end with a 2-character suffix. Figure 7-10 shows a series of valid Motorola data records.

**Figure 7-10**  
An Example of Motorola EXORciser Format



Each data record begins with the start characters S1. The third and fourth characters represent the byte count, which expresses the number of data, address and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix. Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be three less than the byte count. The suffix is a 2-character checksum, which equals the one's complement of the binary summation of the byte count, address and data bytes.

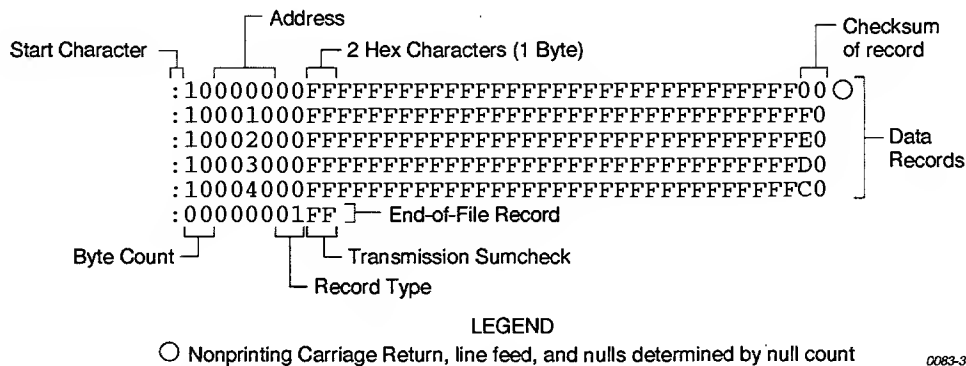
The end-of-file record consists of the start characters S9, the byte count, the address (in hex) and a checksum. The maximum record length is 250 data bytes.

## Intel Intellec 8/MDS Format, Code 83

Intel data records begin with a 9-character prefix and end with a 2-character suffix. The byte count must equal the number of data bytes in the record.

Figure 7-11 simulates a series of valid data records. Each record begins with a colon, which is followed by a 2-character byte count. The 4 digits following the byte count give the address of the first data byte. Each data byte is represented by 2 hexadecimal digits; the number of data bytes in each record must equal the byte count. Following the data bytes of each record is the checksum, the two's complement (in binary) of the preceding bytes (including the byte count, address, record type and data bytes), expressed in hex.

**Figure 7-11**  
An Example of Intel Intellec 8/MDS Format

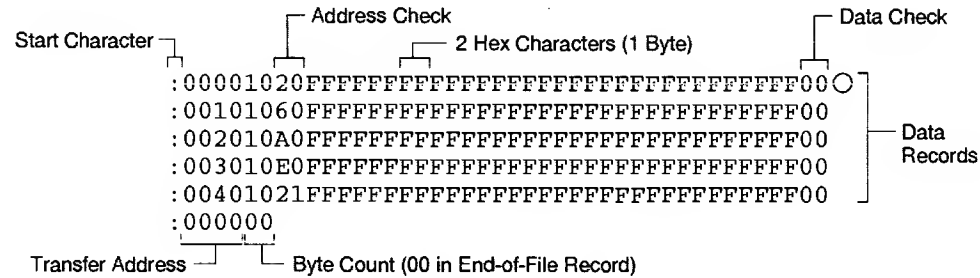


The end-of-file record consists of the colon start character, the byte count (equal to 00), the address, the record type (equal to 01) and the checksum of the record.

# Signetics Absolute Object Format, Code 85

Figure 7-12 shows the specifications of Signetics format files. The data in each record is sandwiched between a 9-character prefix and a 2-character suffix.

Figure 7-12  
An Example of Signetics Absolute Object Format



LEGEND  
○ Nonprinting Carriage Return, line feeds, and nulls determined by null count 0084-2

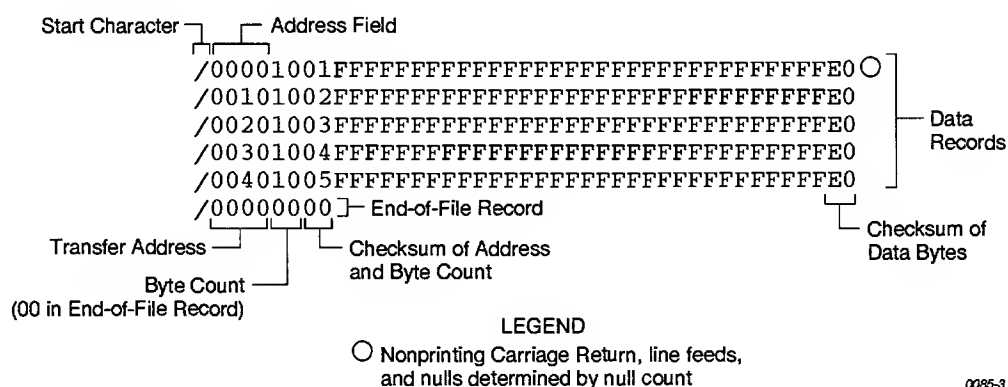
The start character is a colon. This is followed by the address, the byte count, and a 2-digit address check. The address check is calculated by exclusive ORing every byte with the previous one, then rotating left one bit. Data is represented by pairs of hexadecimal characters. The byte count must equal the number of data bytes in the record. The suffix is a 2-character data check, calculated using the same operations described for the address check.

The end-of-file record consists of the colon start character, the address and the byte count (equal to 00).

## Tektronix Hexadecimal Format, Code 86

Figure 7-13 illustrates a valid Tektronix data file. The data in each record is sandwiched between the start character (a slash) and a 2-character checksum. Following the start character, the next 4 characters of the prefix express the address of the first data byte. The address is followed by a byte count, which represents the number of data bytes in the record, and by a checksum of the address and byte count. Data bytes follow, represented by pairs of hexadecimal characters. Succeeding the data bytes is their checksum, an 8-bit sum, modulo 256, of the 4-bit hexadecimal values of the digits making up the data bytes. All records are followed by a carriage return.

**Figure 7-13**  
An Example of Tektronix Hex Format



0085-3

Data is output from UniSite starting at the first RAM address and continuing until the number of bytes in the specified block has been transmitted. UniSite divides output data into records prefaced by a start character and an address field for the first byte in the record.

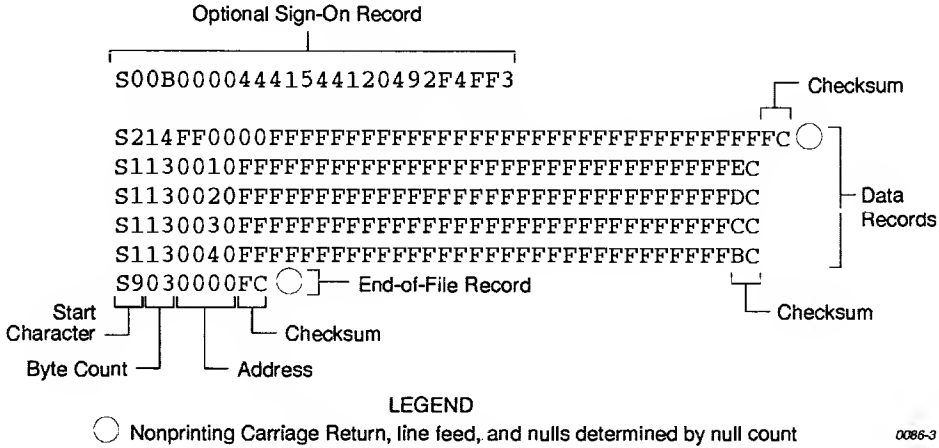
The end-of-file record consists of a start character (slash), followed by the transfer address, the byte count (equal to 00), and the checksum of the transfer address and byte count.

An optional abort record contains 2 start characters (slashes), followed by an arbitrary string of ASCII characters. Any characters between a carriage return and a / are ignored.

# Motorola EXORmacs Format, Code 87

Motorola data files may begin with an optional sign-on record, initiated by the start characters S0. Data records start with an 8- or 10-character prefix and end with a 2-character suffix. Figure 7-14 shows a series of Motorola EXORmacs data records.

**Figure 7-14**  
*An Example of Motorola EXORmacs Format*



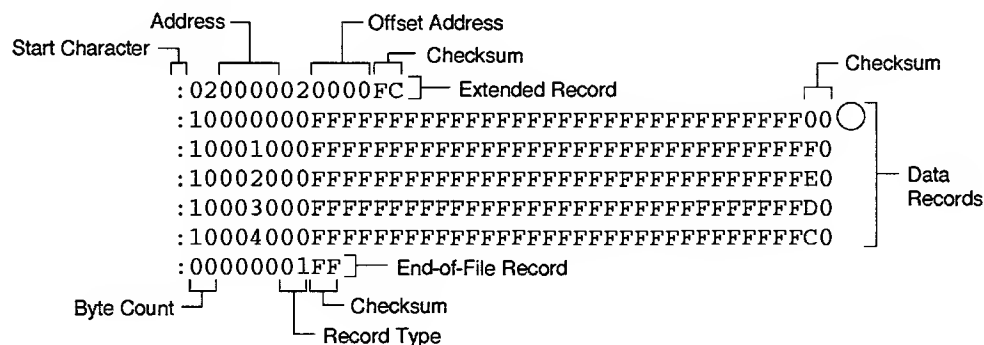
Each data record begins with the start characters S1 or S2; S1 if the following address field has 4 characters, S2 if it has 6 characters. The third and fourth characters represent the byte count, which expresses the number of data, address and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix (6 characters for addresses above hexadecimal FFFF). Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be 3 or 4 less than the byte count. The suffix is a 2-character checksum, the one's complement (in binary) of the preceding bytes in the record, including the byte count, address and data bytes.

The end-of-file record begins with an S9 start character. Following the start characters are the byte count, the address and a checksum. The maximum record length is 250 data bytes.

## Intel MCS-86 Hexadecimal Object, Code 88

The Intel 16-bit Hexadecimal Object file record format has a 9-character (4-field) prefix that defines the start of record, byte count, load address, and record type and a 2-character checksum suffix. Figure 7-15 shows a sample record of this format.

**Figure 7-15**  
*An Example of Intel MCS-86 Hex Object*



### LEGEND

○ Nonprinting Carriage Return, line feed, and nulls determined by null count

0087-4

The four record types are described below.

#### 00 –Data Record

This begins with the colon start character, which is followed by the byte count (in hex notation), the address of the first data byte, and the record type (equal to 00). Following these are the data bytes. The checksum follows the data bytes and is the two's complement (in binary) of the preceding bytes in the record, including the byte count, address, record type and data bytes.

#### 01 –End Record

This end-of-file record also begins with the colon start character. This is followed by the byte count (equal to 00), the address (equal to 0000), the record type (equal to 01) and the checksum, FF.

#### 02 –Extended Segment Address Record

This is added to the offset to determine the absolute destination address. The address field for this record must contain ASCII zeros (Hex 30s). This record type defines bits 4 to 19 of the segment base address; it can appear randomly anywhere within the object file and affects the absolute memory address of subsequent data records in the file. The following example illustrates how the extended segment address is used to determine a byte address.



**Problem:**

Find the address for the first data byte for the following file.

: 02 0000 02 1230 BA  
: 10 0045 00 55AA FF .....BC

**Solution:**

- Step 1. Find the record address for the byte. The first data byte is 55.  
Its record address is 0045 from above.
- Step 2. Find the offset address. The offset address is 1230 from above.
- Step 3. Shift the offset address one place left, then add it to the record address, like this:

1230	Offset address (upper 16 bits)
+ 0045	Record address (lower 16 bits)
<hr/> 12345	20-bit address

The address for the first data byte is 12345.

---

*Note: Always specify the address offset when using this format, even when the offset is zero.*

During output translation, the firmware will force the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.

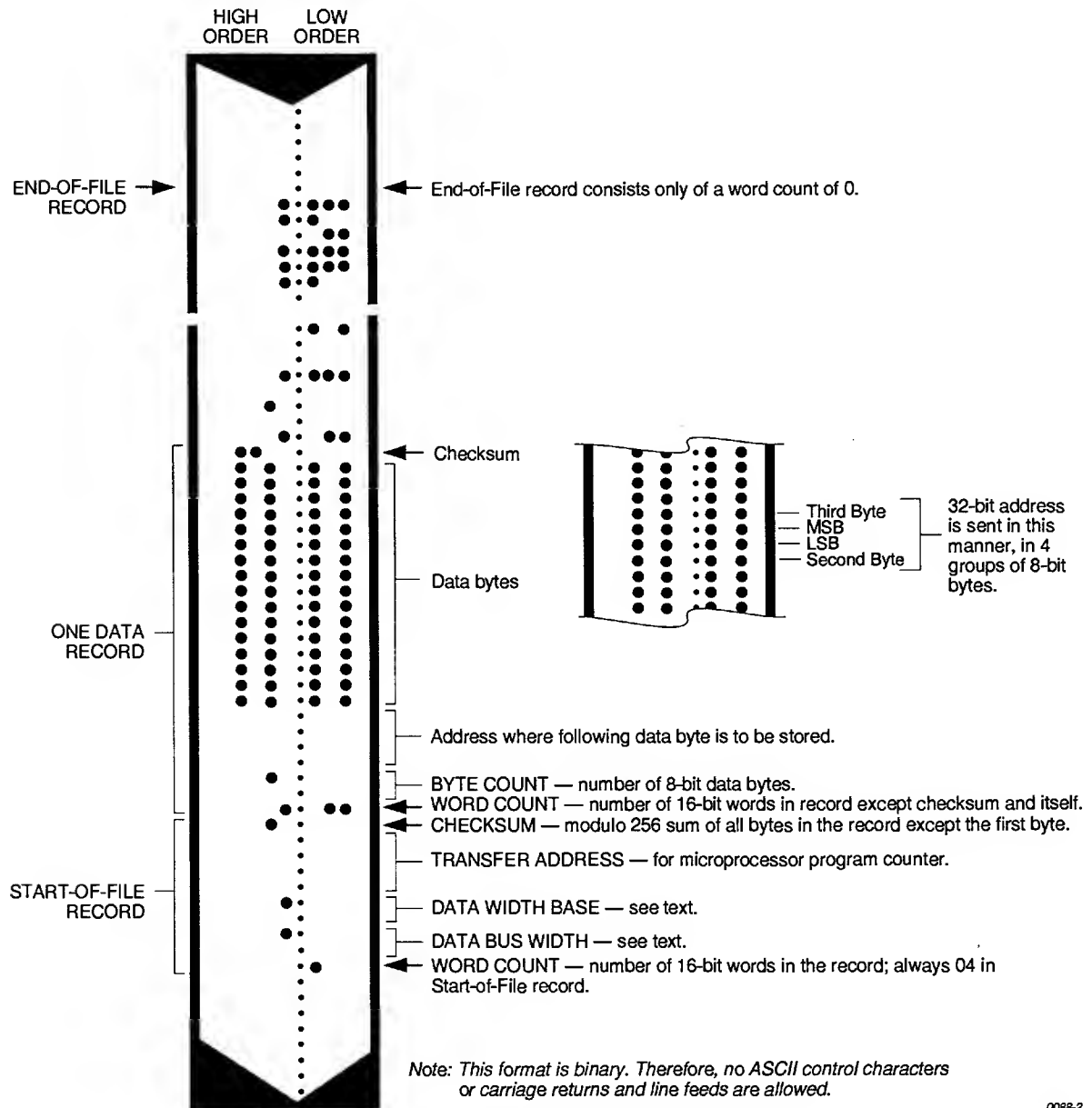
**03 –Start Record**

This record type is not sent during output by Data I/O translator firmware.

## Hewlett-Packard 64000 Absolute Format, Code 89

Hewlett-Packard Absolute is a binary format with control and data-checking characters. See Figure 7-16.

**Figure 7-16**  
An Example of HP 64000 Absolute Format



0088-2

Data files begin with a Start-of-file record, which includes the Data Bus Width, Data Width Base, Transfer Address, and a checksum of the bytes in the record.

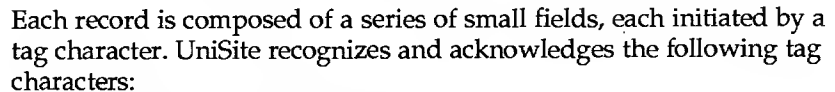
The Data Bus Width represents the width of the target system's bus (in bits). The Data Width Base represents the smallest addressable entity used by the target microprocessor.

The Data Bus Width and Data Width Base are not used by UniSite during download. During upload, the Data Bus Width will be set to the current Data Word Width, and the Data Width Base will be set to 8. The Transfer Address is not used by UniSite.

Data records follow the Start-of-file record. Each begins with 2 byte counts: the first expresses the number of 16-bit bytes in the record not including the checksum and itself; the second expresses the number of 8-bit data bytes in the record. Next comes a 32-bit address, which specifies the storage location of the following data byte. Data bytes follow; after the last data byte is a checksum of every byte in the record except the first byte, which is the word count.

The End-of-file record consists of a one byte word count, which is always zero. Leader and trailer nulls, normally 50 each, are suppressed in this translation format.

**Figure 7-17**  
*An Example of the TI SDSMAC Format*



- The start-of-file record begins with a tag character and a 12-character file header. The first four characters are the byte count of the data bytes; the remaining file header characters are the name of the file and may be any ASCII characters (in hex notation). Next come interspersed address fields and data fields (each with tag characters). If any data fields appear before the first address field in the file, the first of those data fields is assigned to address 0000. Address fields may be expressed for any data byte, but none are required.

Data records follow the same format as the start-of-file record but do not contain a file header. The end-of-file record consists of a colon (:) only. The output translator sends a CTRL-S after the colon.

## JEDEC Format, Codes 91 and 92

### Introduction

The JEDEC (Joint Electron Device Engineering Council) format is used to transfer fuse and test vector data between UniSite and a host computer. Code 91 is full format, and includes all the data fields (such as note and test fields) described on the following pages. Code 92 is the Kernel, or shorter format. The JEDEC Kernel format includes only the minimum information needed for the programming; it does not, for example, include information fields or test vector fields. Prior to transferring a JEDEC file, the appropriate Logic device must be selected.

JEDEC's legal character set consists of all the printable ASCII characters, and four control characters. The four allowable control characters are STX, ETX CR (RETURN) and LF (line feed). Other control characters, such as ESC or BREAK, should not be used.

---

*Note: This is Data I/O Corporation's implementation of JEDEC Standard 3A. For a copy of the strict standard, write:*

*Electronic Industries Association  
Engineering Department  
2001 Eye Street NW  
Washington, D.C. 20006*

### BNF Rules and Standard Definitions

The Backus-Naur Form (BNF) is used in the description here to define the syntax of the JEDEC format. BNF is a shorthand notation that follows these rules:

`::` = denotes "is defined as."

Characters enclosed by single quotes are literals (required).

Angle brackets enclose identifiers.

Square brackets enclose optional items.

Braces { } enclose a repeated item. The item may appear zero or more times.

Vertical bars indicate a choice between items.

Repeat counts are given by a `:n` suffix. For example, a 6-digit number would be defined as

`<number> :: = <digit>:6`

For example, in words, the definition of a person's name reads:

The full name consists of an optional title followed by a first name, a middle name, and a last name. The person may not have a middle name or may have several middle names. The titles consist of: Mr., Mrs., Ms., Miss, and Dr.

The BNF definition for a person's name is:

```
<full name> :: = [<title>] <f. name> {<m.name>} <l. name>
<title> :: = 'Mr.' | 'Mrs.' | 'Ms.' | 'Miss' | 'Dr.'
```

The following standard definitions are used throughout the rest of this document:

```
<digit> :: = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
<hex-digit> :: = <digit> | 'A' | 'B' | 'C' | 'D' | 'E' | 'F'
<binary-digit> :: = '0' | '1'
<number> :: = <digit> {<digit>}
<del> :: = <space> | <carriage return>
<delimiter> :: = <del> {<del>}
<printable character> :: = <ASCII 20 hex ... 7E hex>
<control character> :: = <ASCII 00 hex ... 1F hex> | <ASCII 7F hex>
<STX> :: = <ASCII 02 hex>
<ETX> :: = <ASCII 03 hex>
<carriage return> :: = <ASCII 0D hex>
<line feed> :: = <ASCII 0A hex>
<space> :: = <ASCII 20 hex> | ' '
<valid character> :: = <printable character> | <carriage return> |
<line feed>
<field character> :: = <ASCII 20 hex ... 29 hex> | <ASCII 2B hex ...
7E hex> | <carriage return> | <line feed>
```

**The Design Specification Field**

`<design spec> ::= {<field character>}''`

The first field sent in a JEDEC transmission is the design specification. Both the full and kernel JEDEC formats accept the design specification field. This field is mandatory and it does not have an identifier (such as an asterisk) signaling its beginning. The design specification field consists of general device information. It could, for example, consist of the following information: your name, your company's name, the date, the device name and manufacturer, design revision level, etc. This field is terminated by an asterisk character. Examine the sample transmission shown on the next page of this description -- the first three lines of the file comprise the design specification field. UniSite ignores the contents of this field for downloads and places "Data I/O" in this field for upload operations.

---

*Note: You do not need to send any information in this field if you do not wish to; a blank field, consisting of the terminating asterisk, is a valid design specification field.*

**The Transmission Checksum Field**

`<xmit checksum> ::= <hex digit>:4`

The transmission checksum is the last value sent in a JEDEC transmission. The full JEDEC format requires the transmission checksum. The checksum is a 16-bit value, sent as a 4-digit hex number, and is the sum of all the ASCII characters transmitted between (and including) the STX and ETX. The parity bit is excluded in the calculation of the transmission checksum.

Some computer systems do not allow you to control what characters are sent, especially at the end of a line. You should set up the equipment so that it will accept a dummy value of 0000 as a valid checksum. This zero checksum is a way of disabling the transmission checksum, while still keeping within the JEDEC format rules.

**Figure 7-18**  
*An Example of JEDEC Full Format*

0090-2



## JEDEC Field Syntax

```
<field> ::= [<delimiter>]<field identifier>{<field character>}**

<field identifier> ::= 'A' | 'C' | 'D' | 'F' | 'G' | 'K' | 'L' | 'N' | 'P' |
'Q' | 'R' | 'S' | 'T' | 'V' | 'X'

<reserved identifier> ::= 'B' | 'E' | 'H' | 'I' | 'J' | 'M' | 'O' | 'U' | 'W'
| 'Y' | 'Z'
```

Following the design specification field in a JEDEC transmission can be any number of information fields. Each of the JEDEC fields begins with a character that identifies what type of field it is. Fields are terminated by using an asterisk character. Multiple character identifiers can be used to create sub-fields (i.e., A1, A\$, or AB3). Although not required, you may use carriage returns (CR) and line feeds (LF) to improve readability of the data.

## Field Identifiers

Field identifiers which are currently used in JEDEC transmissions are shown above on the "field identifiers" line. The "reserved identifier" line indicates characters not currently used (reserved for future use as field identifiers). JEDEC field identifiers are defined as follows:

A	Access time	N	Note field
B	*	O	*
C	Checksum field	P	Pin sequence
D	Device type	Q	Value field
E	Electrical data field	R	Resulting vector field
F	Default fuse state field	S	Starting vector
G	Security fuse field	T	Test cycles
H	*	U	User data field
I	*	V	Test vector field
J	*	W	*
K	Fuse list field (hex format)	X	default test condition
L	Fuse list field	Y	*
M	*	Z	*
* Reserved for future use			

## Device Field (D)

Device selection by this field is not supported by UniSite. It has been replaced by the QF and QP fields and the manual selection of devices.

## Fuse Information Fields (L, K, F, C)

```
<fuse information> ::= [<default state>] <fuse list> {<fuse list>} [<fuse
checksum>]

<fuse list> ::= 'L' <number> <delimiter> {<binary-digit> [<delimiter>]}
' * '

<fuse list> ::= 'K' <number> <delimiter> {<hex-digit> [<delimiter>]} '**

<default state> ::= 'F' <binary-digit> ' * '

<fuse checksum> ::= 'C' <hex-digit>:4 ' * '
```

Each fuse of a device is assigned a decimal number and has two possible states: zero, specifying a low-resistance link, or one, specifying a high resistance link. The state of each fuse in the device is given by three fields: the fuse list (L field or K field), the default state (F field), and the fuse checksum (C field).

Fuse states are explicitly defined by either the L field or the K field. The character L begins the L field and is followed by the decimal number of the first fuse for which this field defines a state. The first fuse number is followed by a list of binary values indicating the fuse states.

The information in the K field is the same as that of the L field except that the information is represented by hex characters instead of binary values. This allows more compact representation of the fusemap data. The character K begins the K field and is followed by the decimal number of the first fuse. The fuse data follows the fuse number and is represented by hex characters. Each bit of each hex character represents the state of one fuse, so each hex character represents four fuses. The most significant bit of the first hex character following the fuse number corresponds to the state of that fuse number. The next most significant bit corresponds to the state of the next fuse number, etc. The least significant bit of the first hex character corresponds to the state of the fuse at the location specified by the fuse number plus three.

The K field supports download operations only. The K field is not part of the JEDEC standard, but is supported by Data I/O for fast data transfer. The L and K fields can be any length desired, and any number of L or K fields can be specified. If the state of a fuse is specified more than once, the last state specified replaces all previous ones for that fuse. The F field defines the states of fuses that are not explicitly defined in the L or K fields. If no F field is specified, all fuse states must be defined by L or K fields.

The C field, the fuse information checksum field, is used to detect transmitting and receiving errors. The field contains a 16-bit sum (modulus 65535) computed by adding 8-bit words containing the fuse states for the entire device. The 8-bit words are formed as shown in the following figure. Unused bits in the final 8-bit word are set to zero before the checksum is calculated.

Word 00	msb									lsb
Fuse No.	7	6	5	4	3	2	1	0		
Word 01	msb									lsb
Fuse No.	15	14	13	12	11	10	9	8		
Word 62	msb									lsb
Fuse No.	503	-	-	-	499	498	497	496		

Following is an example of full specification of the L, C, and F fields:

```
F0*L0 01010101* L0008 01010111* L1000 0101*C019E*
```

Following is an alternate way of defining the same fuse states using the K field:

```
F0*K0 55* K0008 57* K1000 5* C019E*
```

Another example, where F and C are not specified:

```
L0200 01101010101010101011
010111010110100010010010010*
```

**The Security Fuse Field (G)**

<security fuse>::='G'<binary-digit>''

The JEDEC G field is used to enable the security fuse of some logic devices. To enable the fuse, send a 1 in the G field:

G1\*

**The Note Field (N)**

<note>::='N'<field characters>''

The note field is used in JEDEC transmission to insert notes or comments. UniSite will ignore this field: it will not be interpreted as data. An example of a note field would be:

N Test Preload\*

**The Value Fields (QF, QP, and QV)**

JEDEC value fields define values or limits for the data file, such as number of fuses. The QF subfield defines the number of fuses in the device. All of the value fields must occur before any device programming or testing fields appear in the data file. Files with ONLY testing fields do not require the QF field and fields with ONLY programming data do not require the QP and QV fields.

The QF subfield tells UniSite how much memory to reserve for fuse data, the number of fuses to set to the default condition, and the number of fuses to include in the fuse checksum. The QP subfield defines the number of pins or test conditions in the test vector, and the QV subfield defines the maximum number of test vectors.

**The P Field**

The P field remaps the device pinout and is used with the V (test vector) field. An asterisk terminates the field. The syntax of the field is as follows:

<pin list>::='P'<pin number>:N''

<pin number>::=<delimiter><number>

The following example shows a P field, V field and the resulting application:

P 1 2 3 4 5 6 14 15 16 17 7 8 9 10 11 12 13 18 19 20 \*

V0001 111000HLHHNNNNNNNNNN\*

V0002 100000HHHLNNNNNNNNNN\*

The result of applying the above P and V fields is that vector 1 will apply 111000 to pins 1 through 6, and HLHH to pins 14 through 17. Pins 7 through 13 and 18 through 20 will not be tested.

**JEDEC U and E Fields**

UniSite supports the optional JEDEC U (user data) and E (electrical data) fields. The U and E fields are described below.

---

*Note: Implementation of the JEDEC U and E fields is not part of the JEDEC-3C (JESD3-C) standard.*

---

**User Data (U Field)**

The U field allows user data fuses that do not affect the logical or electrical functionality of the device to be specified in JEDEC files. For instance, the U field can be used to specify the User Data Signature fuse available in some types of PLD devices because this fuse contains information only (it has no logical or electrical functionality).

---

*Note: To have the JEDEC U field processed correctly, you must select the device before downloading the JEDEC file.*

---

The following guidelines apply to the U field:

- The U field must be included for devices with U fuses.
- Each U-field cell must be explicitly provided if the U field is present.
- The F (default fuse state) field does not affect U fuses.
- There can only be one U field in a JEDEC file.
- The U field fuses must be listed in the order they appear in the device.
- The U field must be listed after the L field and E fields (if used), and before the V (test vector) field (if used).
- The U field is specified using binary numbers, since the full number of U-field cells is otherwise unknown.
- The number of cells specified in the U field is not included in the QF (number of fuses) field.
- The U-field cells are not included in the C (fuse checksum) field.
- The U field reads left to right to be consistent with the L (fuse list) and E fields.

The syntax for the U field is as follows:

```
<User Data Fuse List>::'U'<binary-digit(s)>'**'
```

The character U begins the U field and is followed by one binary digit for each U fuse. Each binary digit indicates one of two possible states (zero, specifying a low-resistance link, or one, specifying a high-resistance link) for each fuse.

For example,

```
QF24*
L0000
101011000000000000000000*
E10100111*
C011A*
U10110110*
```

The U field must be specified after the L field.

If the E field is used, the U field must be specified after it.

## Electrical Data (E field)

The E field allows special feature fuses that do not affect the logic function of the device to be specified in JEDEC files.

The following guidelines apply to the E field:

- Each E-field cell must be explicitly provided if the E field is present.
- The F (default fuse state) field does not affect E fuses.
- There can only be one E field in a JEDEC file.
- The E field fuses must be listed in the order they appear in the device.
- The E field must be listed before the C (checksum) field. If the U field is used, the E field must come before the U (user data) field.
- The E field is specified using binary numbers, since the full number of E-field cells is otherwise unknown.
- The number of cells specified in the E field is not included in the QF (number of fuses) field.
- The E-field cells are included in the C (fuse checksum) field.
- The E field reads left to right for the purpose of checksum calculation.

The syntax for the E field is as follows:

```
<Electrical Data Fuse List>::'E'<binary digit(s)>'*
```

The character E begins the E field and is followed by one binary digit for each E fuse. Each binary digit indicates one of two possible states (zero, specifying a low-resistance link, or one, specifying a high-resistance link) for each fuse. For example,

```
QF24*
L0000
10101100000000000000000000000000*
E10100111*           The E field must come before the C
C011A*                field.
U10110110*           If the U field is used, the E field
                        must be specified before it.
```

**Test Field (V field)**

<function test> :: = [<pin list>] <test vector> {<test vector>}

<pin number> :: = <delimiter> <number>

N :: = number of pins on device

<test vector> :: = 'V' <number> <delimiter> <test condition> :N '\*'

<test condition> :: = <digit> 'B' | 'C' | 'D' | 'F' | 'H' | 'K' | 'L' | 'N' |  
'P' | 'U' | 'X' | 'Z'

<reserved condition> :: = 'A' | 'E' | 'G' | 'T' | 'J' | 'M' | 'O' | 'Q' | 'R'  
| 'S' | 'V' | 'W' | 'Y' | 'Z'

Functional test information is specified by test vectors containing test conditions for each device pin. Each test vector contains *n* test conditions where *n* is the number of pins on the device. The following table lists the conditions that can be specified for device pins.

When using structured test vectors to check your logic design, do NOT use 101 or 010 transitions as tests for clock pins: use C, K, U, or D instead.

**Test Conditions**

- 0 Drive input low
- 1 Drive input high
- 2-9 Drive input to supervoltage #2-9
- B Buried register preload (not supported)
- C Drive input low, high, low
- D Drive input low, fast slew
- F Float input or output
- H Test output high
- K Drive input high, low, high
- L Verifies that the specified output pin is low
- N Power pins and outputs not tested
- P Preload registers
- U Drive input high, fast slew
- X Output not tested, input default level
- Z Test input or output for high impedance

---

*Note: C, K, U, D are clocking functions that allow for setup time.*

The C, K, U, and D driving signals are presented after the other inputs are stable. The L, H, and Z tests are performed after all inputs have stabilized, including C, K, U, and D.

Test vectors are numbered by following the V character with a number. The vectors are applied in numerical order. If the same numbered vector is specified more than one time, the data in the last vector replaces any data contained in previous vectors with that number.

The following example uses the V field to specify functional test information for a device:

```
V0001 C01010101NHLLLHHLHLN *
V0002 C01011111NHLLHLLLHLN *
V0003 C10010111NZZZZZZZZZN *
V0004 C01010100NFLHHLFFLLN *
```

## JEDEC Kernel Mode, Code 92

<kernel>::=<STX><design spec><min. fuse information><ETX><xmit checksum>

<design spec>::={<field character>}\*

<min. fuse information>::=<fuse list>{<fuse list>}

You may use the JEDEC kernel format if you wish only to send the minimum data necessary to program the logic device, for example, if you do not want to send any test vectors. If you specify format code 92, UniSite will ignore everything except the design specification field and the fuse information field. The following fields will be ignored if format 92 is specified: C, F, G, Q, V, and X. Also, the security fuse will be set to zero and the transmission checksum will be ignored.

Figure 7-19 shows an example of a kernel JEDEC transmission.

**Figure 7-19**  
*An Example of JEDEC Kernel  
Mode Format*

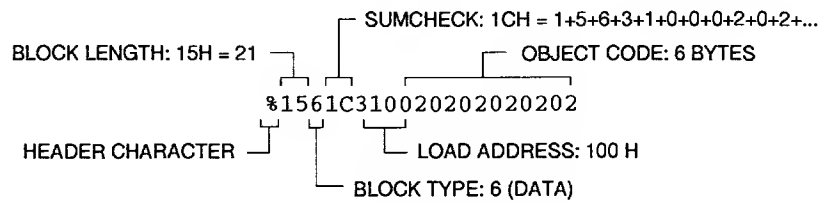
```
<STX>
Acme Logic Design  Jane Engineer   Feb. 29 1983
Widget Decode  756-AB-3456 Rev C Device Mullard 12AX7*
L0000 1111111011 1111111111 1111000000 0000000000
      0000000000 0000000000 0000000000 0000000000
      0000000000 0000000101 1111111111 1111111111
      0000000000 0000000000 0000111101 1111111111
      1111111111 1111110111 1111111111 1111111111*
L0200 1110101111 1111110000 0000000000 0000000000
      1111111111 1111011011 1111111111 1111111110
      0111111111 1111111111 1111111110 1111111111
      1111111111 1111101111 1111111111 1111101111
      0000000000 0000000000 0000*
<EXT>0000
```

0091-2

## Extended Tektronix Hexadecimal Format, Code 94

The Extended Tektronix Hexadecimal format has three types of records: data, symbol and termination records. The data record contains the object code. Information about a program section is contained in the symbol record (UniSite ignores symbol records) and the termination record signifies the end of a module. The data record (see sample below) contains a header field, a load address and the object code. Figure 7-20 lists the information contained in the header field.

**Figure 7-20**  
An Example of Tektronix  
Extended Format



0092-2

Item	No. of ASCII Characters	Description
%	1	Signifies that the record is the Extended Tek Hex format.
Block length	2	Number of characters in the record, minus the %.
Block type	1	6 = data record 3 = symbol record (ignored by UniSite) 8 = termination record
Checksum	2	A 2-digit hex sum, modulo 256, of all the values in the record except the % and the checksum.

### Character Values for Checksum Computation

The number of fields in the file will vary, depending on whether a data or a termination block is sent. Both data and termination blocks have a 6-character header and a 2-to-17 character address.

Character(s)	Value (decimal)	Character(s)	Value (decimal)
0 . . 9	0 . . 9	. (period)	38
A . . Z	10 . . 35	_(underline)	39
\$	36	a . . z	40 . . 65
%	37		

The load address determines where the object code will be located. This is a variable length number that may contain up to 17 characters. The first number determines the address length, with a zero signifying a length of 16. The remaining characters of the data record contain the object code, 2 characters per byte.

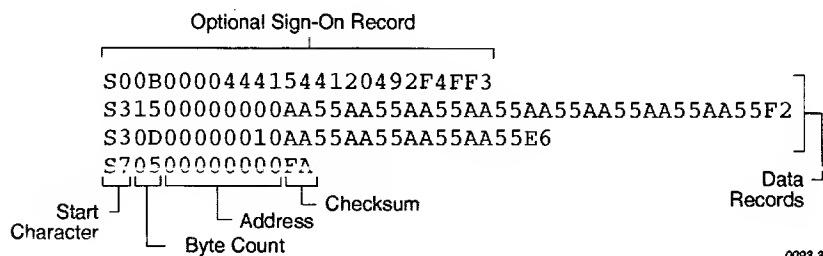
When you copy data to the port or to RAM, set the high-order address if the low-order is not at the default value.



## Motorola 32-Bit Format, Code 95

The Motorola 32-bit format closely resembles the Motorola EXORmacs format, the main difference being the addition of the S3 and S7 start characters. The S3 character is used to begin a record containing a 4-byte address. The S7 character is a termination record for a block of S3 records. The address field for an S7 record may optionally contain the 4-byte instruction address that identifies where control is to be passed and is ignored by UniSite. Figure 7-21 shows a sample of the Motorola 32-bit format.

**Figure 7-21**  
An Example of Motorola  
S3 Format



Motorola data files may begin with an optional sign-on record, initiated by the start characters S0. Data records start with an 8- or 10-character prefix and end with a 2-character suffix.

Each data record begins with the start characters S1, S2 or S3. S1 if the following address field has 4 characters, S2 if it has 6 characters, S3 if it has 8 characters. The third and fourth characters represent the byte count, which expresses the number of data, address and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix (6 characters for addresses above hexadecimal FFFF and 8 characters for addresses above hexadecimal FFFFFFFF). Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be 3, 4 or 5 less than the byte count. The suffix is a 2-character checksum, the one's complement (in binary) of the preceding bytes in the record, including the byte count, address and data bytes.

The end-of-file record begins with an S9 start character. Following the start characters are the byte count, the address and a checksum. The maximum record length is 250 data bytes.

---

## Hewlett-Packard UNIX Format, Code 96

This format divides the data file into data records; each with a maximum size of 250 bytes not including header information. An ID header is added to the beginning of the first record. Each subsequent record has its own header section. The section at the beginning of the file contains the following elements: the header 8004, filename, byte count for the processor information record, and the processor information record.

The header 8004 identifies the type of file being transferred. The first byte of this header (80) indicates that this file is binary and the 04 indicates the type of file (absolute).

The ID header is followed by a 16-byte filename (not used by UniSite).

Next is the byte count, which indicates the size (minus one) of the Processor Information Record that follows. The Processor Information Record is divided into the following data words: Data Bus Width, Data Width Base, Transfer Address LS (least significant), and Transfer Address MS (most significant).

The Data Bus Width represents the width of the target system's bus (in bits). The Data Width Base represents the smallest addressable entity used by the target microprocessor.

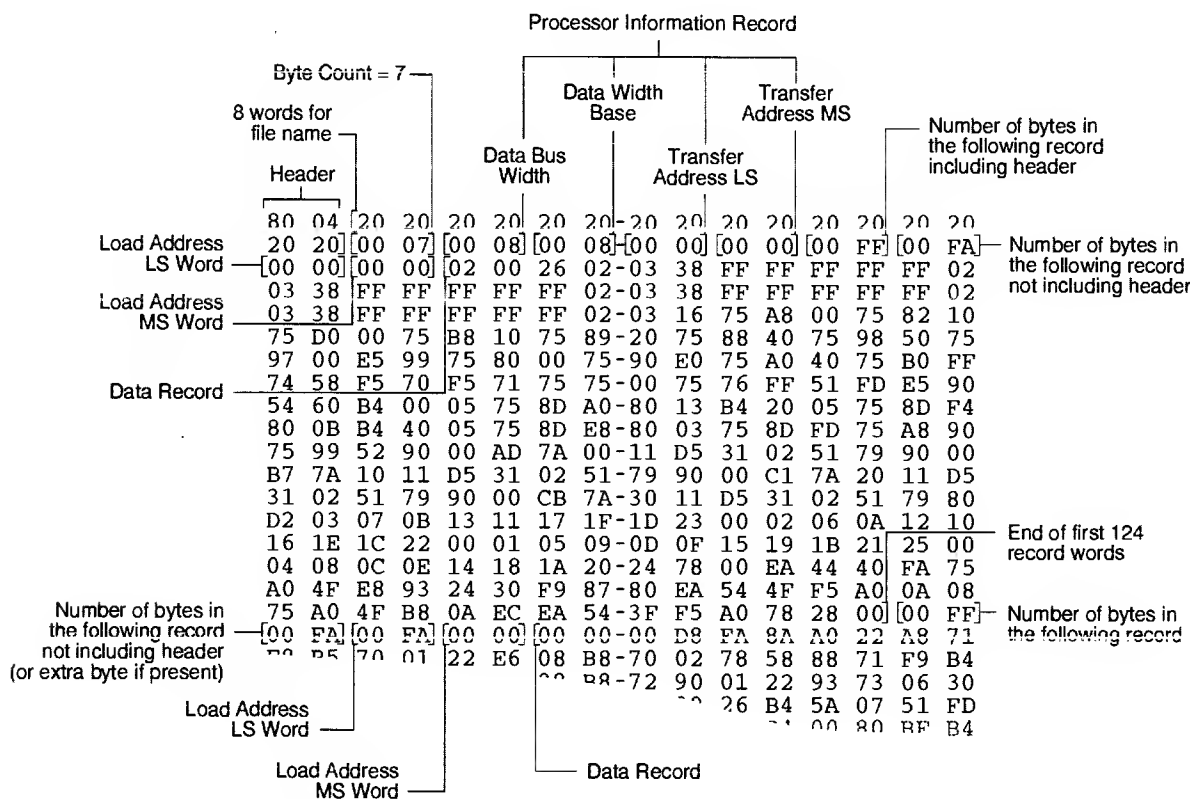
The Data Bus Width and Data Width Base are not used by UniSite during download. During upload, the Data Bus Width will be set to the current Data Word Width, and the Data Width Base will be set to 8. The Transfer Address LS and Transfer Address MS are not used by UniSite.

The data records consist of a header (8 bytes) and the data bytes. The first 2 bytes of the header indicate the size of the data record including the header (minus one). If the number of data bytes in the data record (not including the header) is odd, one extra byte will be added to the data record to ensure that an even number of data bytes exist in the data record. The maximum value for this field is 00FF hex. The next two bytes indicate the number of actual data bytes in the record, not including the header bytes and the extra byte (if present). The maximum value for this field is 00FA hex. The 4 bytes that follow represent the destination address for the data in this record. The rest of the bytes in the record are the data bytes.

This format has no end of file identifier.

The record length during upload is not affected by the upload record size parameter in the Configure/Edit/Communication screen. It is automatically set to transfer records using the maximum size (250 bytes) except for the last record. The size of the last record will be set according to the remaining number of data bytes.

**Figure 7-22**  
**Hewlett-Packard 64000 Unix Format**



This data translation format was generated by a "dump utility" for illustrative purposes. Actual data files are in binary code and are typically generated by the appropriate development software.

0474-2

---

## **Intel OMF386 Format, Code 97**

This data translation format is considered, by Intel, to be proprietary information. Contact your local Intel representative or call (408) 987-8080 for information about the structure of this format.

## Intel OMF286 Format, Code 98

The Intel OMF286 format is a dynamically allocatable file format.

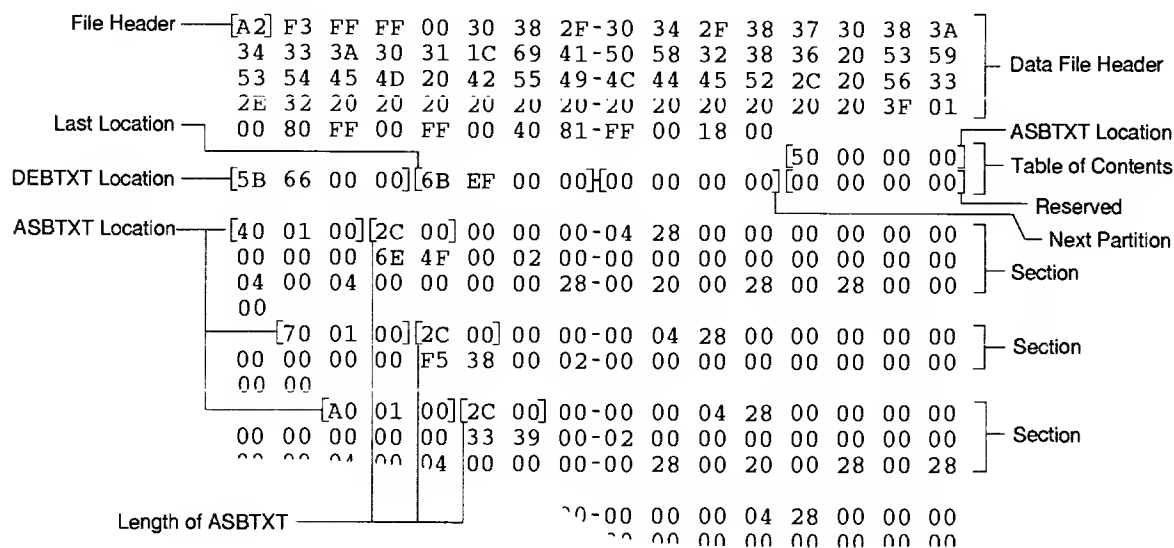
This format has three basic parts: the file header, data file module, and a 1-byte checksum. The file header is hexadecimal number (A2) that identifies this file as an Intel OMF 286 format file. See Figure 7-23.

The first 75 bytes of the data file module is the data file header. The header information is generated and used by the development system and is not used by UniSite, although some characters must fill those bytes. The rest of the data file module consists of one partition.

The partition begins with a 20 byte table of contents. The table of contents specifies the locations of ABSTXT (absolute text), DEBTEXT (debug text), the last location of this partition, and the location of the next partition. The OMF286 format consists of only one partition so this field will be zeros. The rest of the partition consists of sections. The actual data is located in the sections. The first 3 bytes in each section specifies the real address of the text; the next 2 bytes state the length of the text and the remainder of the section is the text (or data). Following the final section of the final partition is a 1-byte checksum representing the complement of the sum of all the bytes in the file including the header. The sum of the checksum byte and the calculated checksum for the file should equal zero. UniSite ignores this checksum.

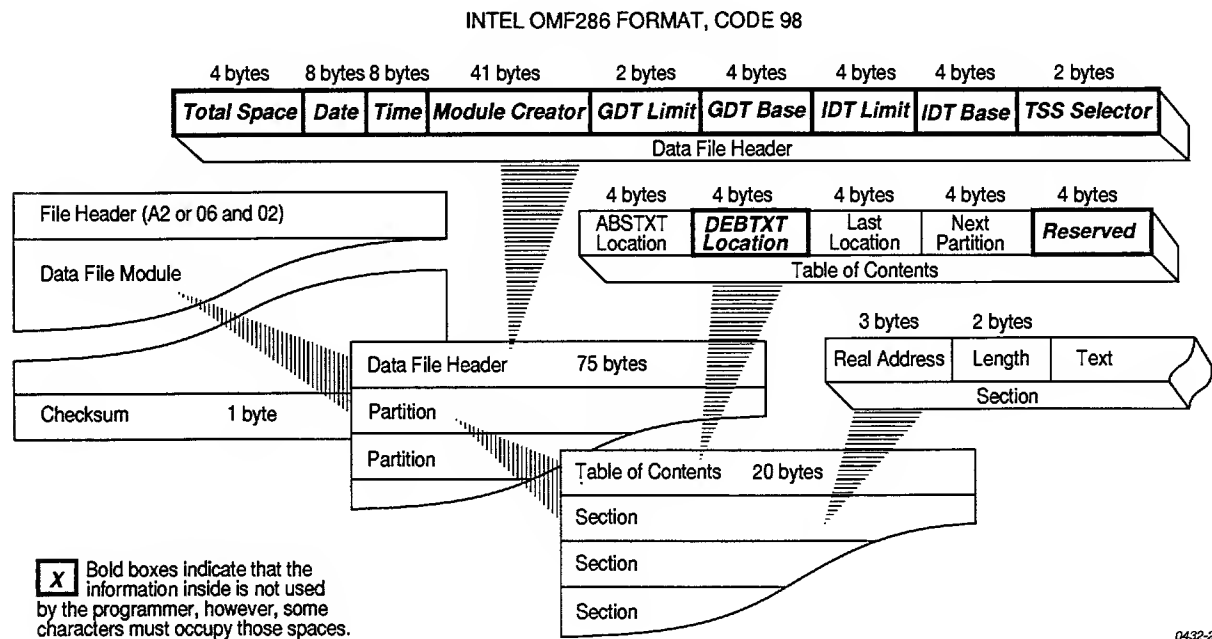
Figure 7-23

A Sample of the Intel OMF286 Format



0431-2

**Figure 7-24**  
A Close-up of the Intel OMF286 Format

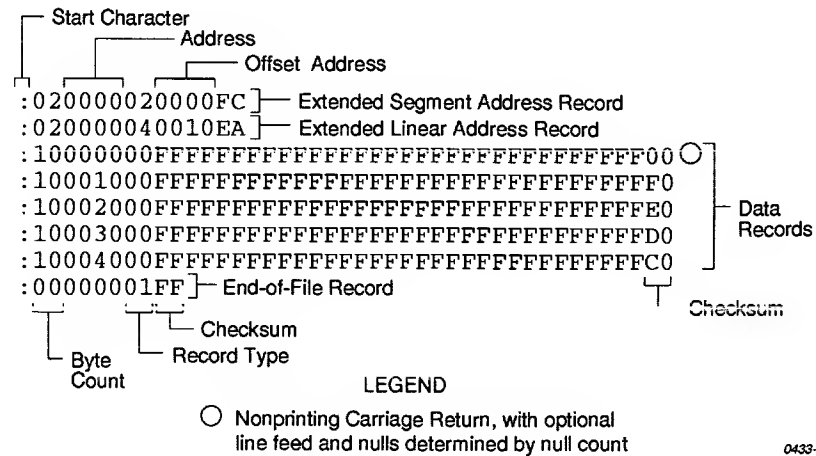


0432-2

## Intel Hex-32, Code 99

The Intel 32-bit Hexadecimal Object file record format has a 9-character (4-field) prefix that defines the start of record, byte count, load address, and record type and a 2-character checksum suffix. Figure 7-25 illustrates the sample records of this format.

**Figure 7-25**  
An Example of the Intel  
Hex-32 Format



The six record types are described below.

### 00 –Data Record

This record begins with the colon start character, which is followed by the byte count (in hex notation), the address of the first data byte, and the record type (equal to 00). Following these are the data bytes. The checksum follows the data bytes and is the two's complement (in binary) of the preceding bytes in the record, including the byte count, address, record type and data bytes.

### 01 –End Record

This end-of-file record also begins with the colon start character and is followed by the byte count (equal to 00), the address (equal to 0000), the record type (equal to 01) and the checksum, FF.

### 02 –Extended Segment Address Record

This is added to the offset to determine the absolute destination address. The address field for this record must contain ASCII zeros (16 hex 30s). This record type defines bits 4 to 19 of the segment base address; it can appear randomly anywhere within the object file and affects the absolute memory address of subsequent data records in the file. The following example illustrates how the extended segment address is used to determine a byte address.

**Problem**

Find the address for the first data byte for the following file.

```
:02 0000 04 0010 EA
:02 0000 02 1230 BA
:10 0045 00 55AA FF ..... BC
```

**Solution:**

- Step 1. Find the extended linear address offset for the data record (0010 in the example).
- Step 2. Find the extended segment address offset for the data record (1230 in the example).
- Step 3. Find the address offset for the data from the data record (0045 in the example).
- Step 4. Calculate the absolute address for the first byte of the data record as follows:

00100000	Linear address offset, shifted left 16 bits
+ 12300	Segment address offset, shifted left 4 bits
+ 0045	Address offset from data record
<u>00112345</u>	32-bit address for first data byte

The address for the first data byte is 112345.

---

*Note: Always specify the address offset when using this format, even when the offset is zero.*

During output translation, the firmware will force the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.

**03 –Start Segment Address Record**

This record, which specifies bits 4-19 of the execution start address for the object file, is not used by UniSite.

**04 –Extended Linear Address Record**

This record specifies bits 16-31 of the destination address for the data records that follow. It is added to the offset to determine the absolute destination address and can appear randomly anywhere within the object file. The address field for this record must contain ASCII zeros (Hex 30s).

**05 –Start Linear Address Record**

This record, which specifies bits 16-31 of the execution start address for the object file, is not used by UniSite.



## Highest I/O Addresses

The following table shows the highest I/O addresses accepted for each Data Translation Format.

Format Number	Format Name	Highest Address (hex bytes)
01-03	ASCII (BNPF, BHLF, and B10F)	N/A
04	Texas Instruments SDSMAC (320)	1FFFF (FFFF words)
05-07	ASCII (BNPF, BHLF, and B10F)	N/A
08-09	5-level BNPF	N/A
10	Formatted Binary	N/A
11	DEC Binary	N/A
12-13	Spectrum	270F
14	POF	N/A
16	Absolute Binary	N/A
17	LOF	N/A
30-32	ASCII-Octal (Space, Percent, and Apostrophe)	3FFFF (777777 octal)
35-37	ASCII-Octal (Space, Percent, and SMS)	3FFFF (777777 octal)
50-52	ASCII-Hex (Space, Percent, and Apostrophe)	FFFF
55-58	ASCII-Hex (Space, Percent, SMS, and Comma)	FFFF
70	RCA Cosmac	FFFF
80	Fairchild Fairbug	FFFF
81	MOS Technology	FFFF
82	Motorola Exorciser	FFFF
83	Intel Intellec 8/MDS	FFFF
85	Signetics Absolute Object	FFFF
86	Tektronix Hexadecimal	FFFF
87	Motorola Exormax	FFFFFFF
88	Intel MCS-86 Hex Object	FFFFFFF
89	Hewlett-Packard 64000 Absolute	FFFFFFF
90	Texas Instruments SDSMAC	FFFF
91-92	Jedec (Full and Kernel)	N/A
94	Tektronix Hexadecimal Extended	FFFFFFF
95	Motorola 32 bit (S3 record)	FFFFFFF
96	Hewlett-Packard UNIX Format	FFFFFFF
97	Intel OMF 386	FFFFFFF
98	Intel OMF 286	FFFFFFF
99	Intel Hex-32	FFFFFFF

# 8 Messages

---

This chapter lists and describes UniSite's system and error messages. If a message requires action, instructions for that action are included in the message's description. Messages are listed in alphabetical order.

Some system and error messages were not documented in this chapter because UniSite provides online help for these messages. To access the online help for a message, press **F3** or **?** when the message appears.

---

*Note: PSM (Package Specific Module) and FSM (Function Specific Module) refer to the modules that are installed on UniSite's front panel. The PSM is the small module located on the left side of the top panel; the FSM is the optional, large module on the right.*

---

## Message List

0 div err

UniSite has experienced a divide-by-zero error that it cannot recover from. This is a fatal error; turn UniSite off and reboot the system. If the error recurs, contact Data I/O Customer Support.

Addr err

UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>Address out of range</b>	The address you tried to select is beyond the selected device's range. Select an address that is within the limits of the device or select a different device. This message appears while you are in the memory editor, fuse editor, or using the under/overflow feature.
<b>Altera POF translator must be selected for POF devices</b>	This message appears when you have selected an Altera POF device and attempt a data transfer operation, such as a download, and have not selected the POF data translation format. Select the POF format as the data translation format and try the operation again.
<b>ASCII entry not allowed in 4-bit mode</b>	This message appears in the memory editor when attempting to go into ASCII entry mode when a 4-bit device is selected. Re-select device or edit in hex mode only.
<b>Beginning of file</b>	This message appears when you are viewing the first block of data and press <b>Ctrl</b> + <b>P</b> (previous page) when you are using the memory editor, vector editor, fuse editor, or the under/overflow feature.
<b>Begin address too large</b>	The beginning address you selected in the memory editor was too large and is beyond the limits of the selected device. Change the begin address to one within the device's range.
<b>Booting non-system disk. Insert system disk. Type ESC and CTRL W to reboot.</b>	This message will appear if UniSite detects a disk other than the System disk installed in drive A during power up. Insert the System disk.
<b>Bus err</b>	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system. If the error recurs, contact Data I/O Customer Support.
<b>Bytes copied = nnnnnn</b>	This message appears while the Copy File operation is in progress; <i>nnnnnn</i> refers to the number of bytes copied.
<b>Cannot access system disk</b>	This message indicates a non-system disk is installed. Make sure the Algorithm disk and System disk are installed.
<b>Calculating sumcheck</b>	This message appears when you are using the Device Check's Sumcheck screen, informing you that the RAM sumcheck is being calculated.
<b>[Computer Remote Control: enter Control-Z to exit.]</b>	This message informs you that UniSite is now in remote control mode and all programmer commands are now read from the remote port. Typing <b>Ctrl</b> + <b>Z</b> returns control to terminal mode.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>Constant over-current fault</b>	This message indicates that an over-current condition exists and UniSite is unable to clear the condition. The over-current could be caused by a hardware failure in UniSite. Reboot the system. If the condition persists, contact Data I/O Customer Support Office.
<b>Constructing Job File Directory</b>	The job files are now being read, in order to put together a job file directory. You can then select one of the files for playback.
<b>Copying file1.ext to file2.ext. Bytes copied = xxxx</b>	This message appears during a Copy operation if you are using the wildcard designation. xxxx in the display is the number of bytes copied into the destination file.
<b>Copying sectors ssss - ssss+120 Reading source disk</b>	This message appears while the Disk Copy command is proceeding. The information presented in this message displays the number of sectors copied in each pass. There are 1440 sectors on each disk. This message is accompanied by the message "Copying sectors ssss - ssss Writing destination disk" which appears while UniSite is writing data onto the destination disk.
<b>Copying sectors ssss - ssss+120 Writing destination disk</b>	This message appears during the Disk Copy routine, indicating that the data is being copied.
<b>Could not initialize default system parameters from disk</b>	When UniSite was booting up, the default and programming system parameters could not be loaded. Reboot UniSite with a different system disk, or contact Data I/O Customer Support for assistance.
<b>Data transfer complete</b>	This message appears after a data transfer with an external source was successfully completed.
<b>Data transfer complete. Data Sum = ssssssss</b>	After a data transfer, this message appears. The data sum represents the calculated sumcheck for the data bytes transferred.
<b>Data transfer complete. Data Sum = ssss. Xmit = ssss.</b>	After a data transfer of a JEDEC file, this message appears. The data sum represents the calculated checksum for the data bytes in the fusemap section of the data transferred. The Xmit sum represents the calculated checksum for all the bytes transferred.
<b>Data transfer complete. Data Sum = ssss. POF CRC = ssss.</b>	After a data transfer of a POF file, this message appears. The data sum represents the calculated checksum for the data bytes transferred. The POF CRC represents the calculated Cyclic Redundancy Check for all the bytes in the POF file up to, but not including, the CRC value.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>Data operation complete: data saved on disk</b>	After a data file is downloaded to disk, this message appears.
<b>Destination file already exists. Hit return to continue, ^Z to abort.</b>	The filename that you have designated as the destination for the data already exists, so existing data will be written over if you execute the operation. This is a precautionary message which occurs on any file operation which could overwrite an existing file.
<b>Disk boot err</b>	UniSite has experienced an error that it cannot recover from. Turn UniSite off and reboot the system. If the problem persists, use another copy of the System disk.
<b>Disk data error</b>	The read or write operation that was attempted could not be completed because there is a problem with the disk; try the operation again with a different disk.
<b>Disk duplication overwrites user RAM. Hit Return to continue, ^Z to abort.</b>	If you copy a disk, UniSite uses User RAM as a buffer. Anything already in User RAM will be overwritten. If you don't want to change User RAM, type <b>Ctrl</b> + <b>Z</b> to halt the disk duplication. Press <b>↵</b> to proceed with the operation.
<b>Disk error, terminal type not saved!</b>	If you try to save the terminal type as one of the power-up parameters and there is a write problem with the disk (the disk is either full or is defective), this message will appear.
<b>Disk open error. Type ESC and Control W to reboot.</b>	This message appears if you try to boot UniSite without the System disk in the disk drive. Insert the System disk in the disk drive and reboot UniSite.
<b>Disk write-protected, terminal type not saved!</b>	If you try to save the terminal type as one of the power-up parameters and the disk is write-protected, this message will appear. Move the write-protect slide so that the hole through the disk is blocked.
<b>Done.</b>	The operation is completed. Proceed to the next operation you want to perform.
<b>Done. Bytes copied = nnnnnn</b>	This message appears after the Copy File operation is complete. It displays the size of the file that was copied in hexadecimal bytes. Proceed to the next operation you want to perform.
<b>End of file</b>	This message appears when you are viewing the last block of data and press <b>Ctrl</b> + <b>N</b> (next page) when you are using the memory editor, vector editor, fuse editor, or the over/under blow feature.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>Fatal system err</b>	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system. If the error recurs, contact Data I/O Customer Support.
<b>FILE ERROR: Can't reach track 0.</b>	If this message appears, a fatal disk error has occurred. The disk drive may be faulty. Contact Data I/O Customer Support.
<b>FILE ERROR: Error in sector preamble.</b>	This error appears when UniSite detects an error with the format of a disk. Use a different disk or reformat the existing disk and try the operation again.
<b>FILE ERROR: No disk in drive.</b>	This message appears when UniSite is trying to access a disk file but the disk drive is empty. Insert the disk with the file to be used.
<b>FILE ERROR: Track not found.</b>	This message appears if UniSite cannot find the disk track associated with the system file, or cannot find the data needed to support whatever action you just requested. If you try the operation again and the error message reappears, a new disk (or a new copy of whatever software or data the unit needs) must be used.
<b>File not initialized! Enter 'C' to initialize, any other key to quit</b>	This message appears within the fuse editor, vector editor or when the under/overflow feature is selected. The file that you have selected is not in a format that is compatible with the feature that you want to use. If you want to use the fuse editor, and the data file you have is not formatted for the device you have selected, typing <b>C</b> reformat the data file to be compatible with the device.
<b>Formatting and initializing user disk.</b>	This message appears while a disk is being formatted.
<b>Hit PF3 or ? to view device specific message</b>	The selected device has specific information associated with it.
<b>Hit return to continue, ^Z to abort.</b>	This message appears after a Verify operation has failed. If you want to ignore the warning and not examine the errors, and proceed with the verify operation, press <b>↓</b> . If you want to investigate the verify errors, press <b>Ctrl</b> + <b>Z</b> and the Verify screen will reappear.
<b>Hit return to switch user menu port, ^Z to abort.</b>	This message is displayed whenever you toggle the User Menu Port parameter. To cancel the port switch operation, press <b>Ctrl</b> + <b>Z</b> , otherwise, press <b>↓</b> to switch the port. The cable between the programmer and the PC (or terminal) should then be moved to the port specified by the parameter.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>IOX init err</b>	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system.
<b>Illegal instr err</b>	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system.
<b>Illegal Key Input: Type control-Z to abort parameter entry.</b>	You pressed a key that is illegal for the field where the cursor is positioned. For example, if you type in a hex number for the Data Word Width field (only decimal numbers are allowed), this message will appear.
<b>Illegal terminal type!</b>	The terminal type number you entered was not one of the choices presented on the Terminal Type screen. Type in a valid terminal type number.
<b>Incompatible User Data Size and Data Word Width.</b>	This message appears when User Data Size is set to an odd value and Data Word Width is greater than 8 bits. Set User Data Size to an even number.
<b>Insert blank device. Hit return.</b>	This message appears during a Quick Copy operation. Remove the newly programmed device or the master device from the device socket, place a blank device in the socket and press <b>↵</b> . UniSite will then begin programming the blank device with RAM data loaded from the master device.
<b>Insert destination disk. Hit return to continue.</b>	This message may appear during the Duplicate Disk or Copy File operation. When this message appears, remove the source disk, insert the destination disk (the disk where you want the data to go to) and press <b>↵</b> . Make sure to use a formatted disk: if you insert an unformatted disk, UniSite will abort the operation and display "Sector not found." If this occurs, perform the Format Disk operation and restart the Duplicate Disk or Copy File operation.
<b>Insert master device. Hit return to continue.</b>	This message appears during a Quick Copy operation. Place the master device into the device socket, lock it into place, and press <b>↵</b> . UniSite will then start to load RAM with data from the master device.
<b>Insert blank device. Hit return to continue or push START lever forward</b>	This message appears during a Quick Copy operation if you are using the SetSite module. Place the blank devices into the SetSite sockets, lock them into place and press <b>↵</b> or move the socket lever to the START position. UniSite will then start to program the parts with data from RAM.
<b>Insert master device. Hit return to continue or push START lever forward</b>	This message appears during a Quick Copy operation if you are using the SetSite module. Place the master device into device socket number one, lock it into place, and press <b>↵</b> or move the socket lever to the START position. UniSite will then start to load RAM with data from the master device.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>Insert source disk. Hit return to continue.</b>	This message appears during a Duplicate Disk operation. When this message appears, remove the destination disk from the disk drive, insert the source disk and press <b>J</b> .
<b>Job file playback ended.</b>	This message informs you that a job file's playback has ended and you may continue with the operation where the job file left off.
<b>Job file save aborted. Keystrokes not recorded.</b>	This message appears in the following situation: If you attempt to end job file recording, and either the system disk is not in the drive or UniSite has difficulty reading the disk, an error message will appear. If you press <b>Ctrl</b> + <b>Z</b> after seeing that message, the above message will appear.
<b>Keystroke recording ended. Select job file for saving.</b>	This message appears after you have pressed <b>Esc Ctrl</b> + <b>J</b> a second time to end recording keystrokes for a job file. Specify a job file number, by typing a number between 0 and 9. Then type in a job file description.
<b>Keystroke recording for job file has begun.</b>	After you press <b>Esc Ctrl</b> + <b>J</b> once, this message will appear. You are now in the job file record mode: every keystroke that you make will be recorded. Type <b>Esc Ctrl</b> + <b>J</b> a second time to end the session.
<b>Loading data from file.</b>	This message appears while data is being loaded into User RAM from a disk's data file.
<b>Loading device algorithm</b>	When you restore a set of system parameters that include a specific device, this message will appear while the programming algorithm is being loaded.
<b>Loading device algorithm file into user RAM.</b>	This message appears when the device programming algorithm is being loaded into User RAM at the first device selection operation after the RAM Device Selection parameter is enabled.
<b>Loading device menu data</b>	This message indicates that UniSite is loading the device and manufacturer selection files.
<b>Loading from disk.</b>	This message appears when UniSite is reading system information or routines from the disk.
<b>Loading programming parameters</b>	When you restore a set of system parameters from the Configuration file directory, this message will appear while the programming parameters are being loaded.



*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>Loop count nnnn = Hit CTRL Z to abort this test</b>	This message appears while a self-test is running in the continuous mode. The loop count <i>nnnn</i> is the number of times the selected test has been repeated.
<b>Memory parity error at:hhhhhh</b>	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system. If the problem persists, record the location at which the error is occurring (represented above by <i>hhhhhh</i> ), and contact Data I/O Customer Support Office.
<b>No disk in drive A.</b>	There is no disk in the disk drive. Insert the System disk into drive A and try the operation again.
<b>Non-blank device. Hit return to continue, ^Z to abort.</b>	This message appears after UniSite has performed a blank check on a device and has detected bits that are not in their erased or blank state, and are not illegal bits. If you press <b>↵</b> , UniSite proceeds with the Programming operation and programs over the existing data. If you press <b>Ctrl</b> + <b>Z</b> , the Program screen will reappear and you can try the operation again with another device. The Blank Check parameter must be enabled before this test can be performed.
<b>Odd Memory Begin Address is not allowed</b>	This message appears when the Memory Begin Address is set to an odd number and you try a device operation on a 16-bit (or larger) device. Set the Memory Begin Address to an even number and retry the device operation.
<b>OPERATION COMPLETE.</b>	The operation you selected has been completed; you may now proceed with other operations.
<b>OPERATION COMPLETE. Device = hhhhhhhh.</b>	This message appears after a successful Compare Electronic ID operation. <i>hhhhhhh</i> represents the device's electronic ID.
<b>OPERATION COMPLETE. Sumcheck = hhhhhhhh</b>	This message appears after the completion of a Program, Load, or Verify operation. <i>hhhhhhh</i> represents the sumcheck of the data that was programmed into the device. "(Vector test not supported)" will be appended to the message when you attempt to perform a structured vector test on devices with more than 84 pins.
<b>OPERATION COMPLETE. Sumcheck = hhhhhhhh. Set Sumcheck = ssssssss</b>	This message appears after the completion of a Set Program, Load, or Verify operation. <i>hhhhhhh</i> is the sumcheck of data that was just programmed into the last set member. <i>sssssss</i> is the sumcheck of all the set members that have been programmed.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>Options installed. Hit Return after changing your terminal settings.</b>	This message appears on the Serial Port Configuration screen after serial port parameters have been changed and <b>↵</b> has been pressed. When this message appears, UniSite suspends screen output until you press <b>↵</b> a second time. Make sure you configure the terminal to match the new settings for the serial port.
<b>Parameter Entered</b>	This message acknowledges that the parameter you entered was accepted.
<b>Parameter Field Full. Hit return or arrows to enter, CTRL Z to abort.</b>	This message appears when you try to enter too many characters into a parameter field. Press <b>↵</b> , <b>F1</b> or <b>F2</b> to enter the parameter.
<b>Power Down</b>	UniSite has experienced a power down condition.
<b>Pre-format check.</b>	<p>This message appears when you have selected the Format Disk operation, and means that UniSite is checking to see if the disk you want to format is a system disk.</p> <p>If this error occurs, go to the self-test screen and re-execute the test(s) that show status of F (Fail). If the test(s) fails while the device socket is empty, UniSite may require service. Contact Data I/O Customer Support.</p>
<b>Purging filename.ext</b>	This message will appear if you are using the wildcard (*) designation to purge more than one file at once; for example, type <b>27*.dat</b> to delete both the files <b>27512.dat</b> and <b>27256.dat</b> .
<b>Reading user data file size</b>	This message appears while UniSite is reading the data file size from disk.
<b>Recording system state parameters.</b>	This message appears after you select a file number for the set of system parameters that you want to save. This message remains until UniSite is finished recording the parameters.
<b>Restoring system state variables.</b>	This message appears while UniSite is reading the recorded system variables from the selected file.
<b>RTC err</b>	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system. If the error recurs, contact Data I/O Customer Support.
<b>RTE init err</b>	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system. If the error recurs, contact Data I/O Customer Support.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

Saving data to file.	This message appears while data is being written to a file on disk.
Saving parameters	This message appears when UniSite is saving the selected variables onto the disk.
Saving job file.	This message appears when UniSite is saving a job file.
Search pattern not found	If you specified a data pattern for a file that does not contain that pattern, this message will appear. This message appears while UniSite is in the memory editor or in the under/overflow display.
Security fuse violation. Hit return to continue, ^Z to abort	This message appears when you try to program an EE device with the security fuse already blown. If you continue by pressing <b>↵</b> , the program operation will be performed and previous data in the device will be overwritten.
System error. Please contact Data I/O.	Contact Data I/O Customer Support.
System parameters restored.	This message appears when you have restored a configuration file from the Restore System Parameters screen.
System parameters saved.	This message appears when you save a set of system parameters.
Task error	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system. If the error recurs, contact Data I/O Customer Support.
Testing	This message appears when a self-test is in progress.
TEST HALTED: Socket not empty, hit return to continue, ^Z to abort.	The self-test that you are attempting requires that the device socket does not contain any devices. Remove the socketed part and try the operation again, or type <b>Ctrl</b> + <b>Z</b> to abort the operation.
<hr/> <b>CAUTION:</b> <i>If you press the carriage return key, UniSite will run the test and the socketed device could be damaged.</i>	
Transferring data.	This message appears while a data transfer operation is being performed.

*Note: Some system and error messages were omitted on purpose. UniSite provides online help for these messages. To access online help for a message, press **F3** or **?** when the message appears.*

<b>[transparent mode]</b>	This message appears on the screen when UniSite enters the transparent mode. To exit transparent mode, type <b>Esc</b> <b>Ctrl</b> + <b>T</b> .
<b>Trc init err</b>	UniSite has experienced an error that it cannot recover from; turn UniSite off and reboot the system. If the error recurs, contact Data I/O Customer Support.
<b>Updating device algorithms</b>	This message appears when the algorithm disk is being updated in the Update Device Algorithms operation. Do not remove the algorithm disk or turn off system power for the duration of the operation.
<b>User RAM sumcheck = ssssssss</b>	This message contains the sumcheck for all of User RAM and is generated in the Sumcheck device check screen. This calculation is done regardless of whether user data is in RAM or on disk.
<b>Using Keep Current algorithm in filename.KCx</b>	This message appears when the "replaced" Keep Current algorithm is used during a normal device selection operation where <i>filename.KCx</i> is the Keep Current algorithm file.
<b>Vector out of range</b>	The vector you tried to select does not exist for the device you have selected. Select a vector that is within the limits of the device or select a different device. This message may appear while you are using the vector editor.
<b>Waiting for self-test completion.</b>	This power-up message shows up only if you are changing the terminal selection before the power-up self-test has been completed.
<b>WARNING Algorithm disk in drive. Hit return to continue, ^Z to abort.</b>	This message will appear if you are attempting a file operation and have the Algorithm disk installed in the disk drive.
<b>WARNING: System disk in drive. Hit return to continue, ^Z to abort.</b>	This message appears during any file operation that displaces disk data. Any information currently on the disk will be erased and is not retrievable. Press <b>Enter</b> to go ahead with the operation. Press <b>Ctrl</b> + <b>Z</b> , to cancel the operation.



# Glossary

---

<b>Action Symbol</b>	Found in the upper left corner of the screen, the action symbol rotates to indicate that the programmer is performing an operation.
<b>Address</b>	A coded instruction designating the location of data or program segments in storage.
<b>Address Offset</b>	A value subtracted from addresses during input translation and then added to addresses during output translation.
<b>Algorithm</b>	The software file containing information to program a specific device, usually contained on a floppy disk.
<b>All Parameters</b>	Also referred to as the Complex Parameters screen, the All Parameters screen displays all applicable parameters on a parameter entry screen.
<b>Approval</b>	Indication that a device manufacturer has tested an algorithm to support a specific device on a programmer. The level of an approval varies by device manufacturer, but an approval usually indicates both yield and waveform analysis.
<b>AutoBaud</b>	A special feature on the programmer that senses the baud rate of equipment connected to the programmer and sets the programmer's baud rate to match the equipment's baud rate. AutoBaud is designed for equipment that cannot support baud rates as high as 9600 baud, such as 1200/2400 baud modems.
<b>Base</b>	The interface between the module (PinSite) and the device. The Base routes the signals between the device and the Universal Pin Drivers inside the programmer.
<b>Baud Rate</b>	A measure of data flow. The number of signal elements per second based on the duration of the shortest element. When each element carries one bit, the Baud rate is numerically equal to bits per second.
<b>BGA</b>	An acronym for Ball Grid Array, a type of device package. Usually a square device with one side populated with small solder balls as leads.

<b>Blank Check</b>	A device check that checks a device for programmed bits. If no programmed bits are found, the device is considered blank.
<b>Block Size</b>	The hexadecimal number of bytes to be transferred in a data transfer. The beginning of the block is defined by a begin address, and the end of the block is the sum of the block size and the begin address minus one.
<b>Byte Swap</b>	See Odd/Even Byte Swap.
<b>Cards</b>	Also called PC Cards are 68-pin, integrated circuit cards. PC Cards usually follow the PCMCIA/JEIDA (PC Memory Card International Association) standards, which govern 68-pin memory or I/O type cards.
<b>Command Window</b>	The left side of the screen. At the top of the window is the menu name, displayed in uppercase letters. Below the menu name, the available commands are displayed in upper- and lowercase characters.
<b>Communications Parameters</b>	The various settings that determine the I/O characteristics of your equipment. The parameters include baud rate, stop bits, data bits, handshaking, etc.
<b>Compare Electronic ID</b>	A command that compares the electronic signature of the socketed device against the electronic signature specified in the currently selected algorithm.
<b>Compensated Vector Test</b>	A device test that enables load compensation on PLD output pins under test during vector testing. This may eliminate structured test errors when testing PLDs sensitive to output loading, where many of the devices register transitions simultaneously.
<b>Complex Parameter Screen</b>	See All Parameters.
<b>Computer Remote Control</b>	A command set that may be used to operate a programmer remotely. These commands are usually the basis for external programmer drivers, which may operate a programmer from a PC or other host. See also Remote Mode.
<b>Context-sensitive</b>	Information that changes depending on the screen position. With the programmer, every time you move the cursor to a different field, the information on the online help screen changes.
<b>Continuity Check</b>	A device check that tests for open device pins before performing a device operation.
<b>CRC</b>	An acronym for Computer Remote Control. See Remote Mode.
<b>Cross Programming</b>	A programming operation that allows a single generic programmable logic device (PLD) to be configured as any one of many PLD architectures. Consequently, the generic device can take on the function of many subset devices. For example, a 16V8 generic PLD can be configured as a 16R4, 16R8, 16L8, etc.

<b>Data Bits</b>	A communication parameter that specifies the number of bits per byte.
<b>Data Representation</b>	The manner in which the data in user memory appears on the screen. You can select either <b>X</b> and <b>-</b> , or <b>0</b> and <b>1</b> , where <b>X</b> and <b>0</b> represent an unprogrammed state, and <b>-</b> and <b>1</b> represent a programmed state.
<b>Data Word Width</b>	The word width of the data to be used during a device operation. For 8-bit (or above) devices, the maximum is 64, and the minimum word width is equal to the device width. For 4-bit devices, the word width can be 4, 8, 16, or 32. This value should match the word width of the data bus in the target system for the device being programmed.
<b>Destination</b>	The place to where you are sending something. The "something" you are sending to the destination is almost always data. The "where" you are sending the data can be RAM, a disk file, or one of the programmer's serial ports.
<b>Device Begin Address</b>	The first hexadecimal address of device data to use for a device operation. If programming, it represents the first address to program. If verifying, it represents the first address to verify.
<b>Device Block Size</b>	The size of device data to be used in device operations.
<b>Device Operation</b>	Usually a term that refers to loading, programming, or verifying. However it can also refer to other available commands, such as device checks and electronic erasing.
<b>Device Word Width</b>	The number of bits in the data word of the device.
<b>DIP</b>	An acronym for Dual In-line Package, a type of device package. <b>SDIP</b> (Shrink Dual Inline Package) is similar to DIP but with more leads at tighter pitch (a separate adapter is required to program SDIPs).
<b>Dialog Window</b>	The largest window on the screen. The dialog window displays different information and system parameters, depending on the selected command.
<b>Download Data</b>	A file operation that moves a data file from a host computer to the programmer's RAM or disk.
<b>Download Echoing</b>	Displays the data being downloaded.
<b>Download Host Command</b>	A command that is sent from the programmer to the host during a download. The command tells the host to begin sending data to the programmer.
<b>E-MICRO</b>	An acronym for Programmable Microcontroller, a type of device technology.
<b>EPROM</b>	An acronym for Erasable Programmable Read-Only Memory. (Usually refers to UV erasable memories.)



<b>EEPROM</b>	An acronym for Electronically Erasable Programmable Read-Only Memory. The device can be either completely or partially erased electrically in circuit or on the programmer.
<b>Electronic ID</b>	The combination of bytes that identify the device number and manufacturer of a programmable device.
<b>Enhanced Security Fuse Capability</b>	Found on EMICROs, the Enhanced Security Fuse Capability allows security fuse data to be stored in a data file. For more information, or to see if a device supports this capability, see the device manufacturers data book.
<b>ESD</b>	An acronym for Electro-Static Discharge.
<b>False Positive</b>	In programming, a misprogrammed fuse that retains minimal operational characteristics so that it passes the fuse test. These may be inadequately programmed, or over-programmed so that they will fail later in circuit.
<b>Family Code</b>	A 2- or 3-digit hexadecimal number that groups devices by their programming algorithms. When combined with the pinout code, the family code identifies the device you are using to the programmer.
<b>Family/Pinout Code</b>	Two- or three-digit codes used by some Data I/O programmers to identify programming variables including pinout, address limit, and programming algorithms. Several similar devices can share the same, valid family/pinout code. And, because similar devices do not always share the same characteristics, some device checks will not be performed if you select a device with family/pinout codes. Thus, if you want specific device checks to be performed, you should select the device by manufacturer/part number.
<b>File Transfer Operations</b>	An operation involving the transfer of data between the programmer and a host. Upload and download are file transfer operations.
<b>Filename</b>	The name of the disk file to use during file operations. The filename must follow standard DOS conventions: up to eight alphanumeric characters, followed by an optional three-character file extension, with the two fields separated by a period. An example of a valid filename would be 27256.dat or filename.c.
<b>Fuse Verification</b>	A type of post-programming device check that checks the fuse pattern programmed into a logic device with the pattern in user memory.
<b>Fusemap</b>	The fuse-level description portion of a programmable integrated circuit. Fusemaps are typically files in JEDEC Standard #3A and are downloaded to PLD programmers for device implementation.
<b>Handshaking</b>	The required sequence of signals for communication between two units. The I/O bus protocol for a unit defines its handshaking requirements. This is especially true for asynchronous I/O systems in which each signal requires a response to complete an I/O operation.

<b>High-speed Download</b>	A special feature of the programmer that allows the programmer to download data from a PC at 115.2K baud. To take advantage of this feature, you must be using HiTerm on a PC, you must enable the High Speed Download parameter, and you must send User Menu Data to the Remote port.
<b>High-speed Logic Drivers</b>	A device test that increases the speed of the logic transitions between 0 to 1 and 1 to 0 of the test vector input states. This test is a diagnostic tool designed to help debug and classify test vector failures. Specifically, this test is designed to help identify vector transitions that are speed dependent.
<b>Host</b>	A micro, mini, or mainframe computer used to control the programmer in Remote mode. You must use a software driver, such as Data I/O's PROMlink to allow the computer to communicate with the programmer.
<b>Host Command (download &amp; upload)</b>	The command that is sent from the programmer to the host system during uploading/downloading. See Download Host Command and Upload Host Command.
<b>I/O Address Offset</b>	This value influences the beginning address where data is stored during a file transfer operation. For uploads, the I/O Offset represents the address to start loading a formatted data file. For downloads, the I/O Offset is subtracted from the beginning address in the formatted data file. The result is then added to the memory begin address to determine where the block of data is loaded.
<b>I/O Timeout</b>	The amount of time that the programmer will wait for a data transfer to begin.
<b>I/O Translation Format</b>	See Translation Formats.
<b>Illegal Bit</b>	An illegal bit occurs when a device contains a programmed location and the data file specifies that the location should be unprogrammed.
<b>Illegal Bit Check</b>	A test that determines whether or not a socketed device contains any illegal bits.
<b>Instrument Control Code</b>	A 1-digit number that signals or controls data transfers. It also implements a form of remote control that provides peripherals with flow control beyond that provided by software handshaking.
<b>JEDEC</b>	Joint Electron Design Engineering Committee: a committee of programmer and semiconductor manufacturers that provide common standards for programmable issues. Examples include: acceptable test characters for PLDs and standard data transfer/programming formats for PLDs. JEDEC Standard #3 is the industry standard for PLD formats.
<b>JEDEC Standard #3A</b>	The standard PLD data translation format, as defined by JEDEC for PLD design software to communicate with PLD programmers. It defines the states of all fuses in the device (the fusemap) and may include test vectors for DEVICE testing.

<b>JEDEC I/O translate DIP/LCC Vectors</b>	A feature on the programmer that translates test vectors for a device from its DIP package to its PLCC/LCC package, allowing for the different pinouts of the two package types.
<b>JLCC</b>	An acronym for J-style Leadless Chip (or Ceramic) Carrier, a type of device package. A device with hooked leads that are "open" at one end (leads are usually on all four sides).
<b>Job File</b>	A sequence of keystrokes that have been stored in a disk file and that can be played back at a later time.
<b>LCA</b>	An acronym for Logic Cell Array.
<b>LCC</b>	An acronym for Leadless Chip Carrier, a type of device package. A 4-sided ceramic package with pads on the underside for surface mount applications.
<b>LED</b>	An acronym for Light Emitting Diode. The programmer has five LEDs: four on the top cover and one on the disk drive.
<b>Load Data</b>	A device operation that moves device data into the programmer. You can load the programmer with data from a device, from the programmer's internal disk drive, or from a serial port (for example, from the Remote port).
<b>Load Device</b>	A device operation that copies data from a master device into User memory.
<b>Logic Verification</b>	After programming a device, you can select test vector verification, fuse verification, or both types of verification.
<b>Master Device</b>	A device that contains data you wish to program into another device. For example, you would load data from a master device and then program that data into a blank device.
<b>MatchBook</b>	A new type of socketing technology that makes handling surface-mount devices easier.
<b>Memory Begin Address</b>	The first address, in hex, of the first byte of data to be used in device operations. If the data source/destination is RAM, the memory begin address is a RAM address. If the data source/destination is disk, the memory begin address is the offset for a disk file.
<b>Message Bar</b>	The fourth line of the screen. The programmer displays system and error messages in the message bar. The action symbol is also located in the message bar.
<b>Next Device</b>	Used during serial set programming, this value specifies the next device in the set. For example, if you are using 8-bit devices and have specified a word width of 16 bits, it will require two devices to store each 16-bit word. Depending on the value entered, the data programmed into the next device will come from either even addresses or odd addresses.

<b>Non-default Parameters</b>	Also referred to as the Simple Parameter screen, the Non-default parameters screen displays a selected group of parameters on a parameter entry screen. To display all the available parameters, select the All Parameters screen, which is also referred to as the Complex Parameters screen.
<b>Odd/even Byte Swap</b>	Used during device operations for 16-bit devices, this option swaps the Most Significant Bytes (MSB) and the Least Significant Bytes (LSB) of 16-bit words. The programmer stores RAM data and disk file data with the convention that the LSB of a 16-bit word resides in the even byte of memory.
<b>Online Help</b>	Available throughout the programmer, the help screens provide you with both general help and context-sensitive help. The Help screen is divided into four sections: the key listing, the general help, the specific help, and the reminder bar.
<b>Output Record Size</b>	The number of data bytes contained in each data record during upload.
<b>Overblow</b>	A condition in which fuses are blown that should not have been.
<b>Overblown Fuse</b>	A fuse that has been over-programmed such that the surrounding area may have been damaged or such that fuse material splatter was created. Splatter (or rattlers) can cause intermittent shorting.
<b>PAL</b>	An acronym for Programmable Array Logic. PALs are devices with programmable AND and fixed OR arrays. This is a slightly different architecture from a PROM or an FPLA. Other examples of PAL-type architectures from other manufacturers include PEEL and GAL.
<b>Parallel Test Vector Application</b>	Use of internal registers to hold and release a full set of test vectors (e.g., 20 for a 10-input 10-output device) at once. In contrast to serial application, parallel does not require accommodations for clocking contention, and parallel better matches in-circuit PLD operation and board test suites.
<b>Part Number</b>	The number on the device. For example if you are using an Intel 27C256, then the part number of the device is 27C256.
<b>Pin-driver</b>	The electric circuit reading or applying voltage and current pulses to the individual pin of a device, for programming or testing. See also Universal Pin Driver.
<b>Pinout Code</b>	A 2- or 3-digit hexadecimal number that helps identify a device by its pinout. When combined with the family code, the pinout code identifies the device you are using to the programmer.
<b>PGA</b>	An acronym for Pin Grid Array, a type of device package. Usually a square device with one side populated with small pins as leads.
<b>PLCC</b>	An acronym for Plastic Leaded Chip Carrier, a device package with J-shaped leads extending from four sides downward, used for surface mount applications.

PLD	An acronym for Programmable Logic Device, a type of programmable integrated circuit. Architectures range from very simple to very complex. Most PLDs contain two levels of logic, an AND array followed by an OR array.
PROM	An acronym for Programmable Read-Only Memory. A device with fixed AND and programmable OR arrays. This is a slightly different architecture from an FPLA or a PAL.
Program	The controlled application of electrical pulses to program specific fuses or cells.
Program Device	A device operation that copies device data into a socketed device. The programming is done according to the programming algorithm selected in the select device stage. The programming operation can also include a verify operation.
Program Security Fuse	A programming parameter that enables/disables the programming of the device's security fuse.
Program Signature	Available on only a few devices, the Program Signature is a user-definable field that allows the user to program data into the program signature array. For example, the Program Signature could contain the revision level or modification date of the data in the remainder of the device.
Programmable Integrated Circuit	One of the four basic categories of ASICs: the other three being gate arrays, standard cells and full custom devices. PICs are ICs that are user configurable. PLDs and PGAs are examples of programmable integrated circuits.
QFP	An acronym for Quad Flat Pack, a type of device package. A square or rectangular device with leads on all four edges (leads can be either straight or gull-wing). <b>BQFP</b> (Bumper Quad Flat Pack) is a QFP with "bumpers" on the corners to prevent damage to leads. <b>CQFP</b> is a Ceramic QFP or a QFP with carrier. <b>Long Lead Carrier QFP</b> is an ACTEL product with long leads attached to a carrier grid for handling ease. <b>PQFP</b> (Plastic carrier QFP) is a QFP with a plastic protective carrier for handling ease. <b>QFPCAR</b> is a QFP in a carrier. <b>TQFP</b> is a thin QFP.
QUIP	An acronym for Quad Inline Package, a type of device package. Similar to DIP, but with staggered long & short leads (leads on long sides).
Reboot	The process of re-initializing the programmer. After rebooting, the programmer is in the same state as if it had just been turned on.
Registered Devices	Devices that contain registers, rather than being combinatorial only. Registered devices are typically used for sequencers and state machine designs. Typical examples are 16R8, 82S159 and 22V10.

<b>Reject Option</b>	A post-programming device check that pulses the programmed device with voltage to see if the device has programmed per specification. The number of times a device is pulsed varies by manufacturer and by the reject option you select.
<b>Reminder Bar</b>	The bottom line of the screen. The reminder bar tells you what function keys are available and what they will do if pressed.
<b>Remote Mode</b>	The programmer is controlled from a host running a driver program. Device data files can be stored on the programmer's disk and on the host.
<b>Security Fuse</b>	A location in a programmable device that, when programmed, secures the device from readback: the data in the device is unreadable.
<b>Security Fuse Data</b>	The actual data to program into the device's security fuse.
<b>Select Device</b>	A procedure that tells the programmer what device you will be using. You can select a device in one of two ways: by entering the family/pinout code, or by selecting the manufacturer and the device part number.
<b>Self-test</b>	A built-in self-diagnosis command that allows you to test various circuits and subsystems in the programmer, verifying proper operation or isolating possible problem areas.
<b>Serial Set</b>	A method of set programming in which the devices of the set are programmed one at a time instead of all at once.
<b>Serial Test Vector Application</b>	The process of applying test vectors in a serial fashion, one input at a time.
<b>Serial Vector Test</b>	A device test that applies test vector input states serially, starting with pin one and stepping through the remaining pins. This test is a diagnostic tool designed to help debug and classify test vector failures. Specifically, this test is designed to isolate test vectors that are sequence dependent.
<b>Set Programming</b>	A type of programming where a large data file is partitioned and programmed into multiple memory devices.
<b>SIMM</b>	An Acronym for Single Inline Memory Module, a type of device package. A rectangular device with leads on one long edge.
<b>Simple Parameter Screen</b>	See Non-default parameters.
<b>SmartPort</b>	A feature of the programmer that automatically detects and adjusts the programmer to the presence of DCE/DTE protocol.
<b>SOIC</b>	An acronym for Small Outline Integrated Circuit, a type of device package. A rectangular device with gull-wing leads on the long sides. SOP is Small Outline Package. SO is Small Outline.

<b>Source</b>	The place from which something comes. The "something" the source is sending is almost always data. The "where" can be RAM, a disk file, or one of the programmer's serial ports.
<b>SSOP</b>	An acronym for Shrink Small Outline Package, a type of device package. A SOP with more leads at higher and finer pitch. Also called TSOP II.
<b>Status Window</b>	The top three lines of the screen. The following information is displayed in the status window: the name of the data file, the amount of user RAM, the version numbers of the algorithm/system disk, the device manufacturer and part number, the family/pinout code, and the data translation format.
<b>Structured Test Vectors</b>	A string of test conditions applied to a PLD in a programmer/tester to stimulate inputs and test outputs to ensure functionality. A test vector is one such string. For instance, 20 characters for a 20-pin PLD, with 10 input signals and 10 expected outputs.
<b>Structured Test Vectors (design)</b>	Structured vectors created by the design engineer to confirm that the design is operating as intended. For instance, that a 10-bit counter is counting to 10. Design vectors are used both in preprogramming simulation and in manufacturing.
<b>Structured Test Vectors (device)</b>	Structured vectors created by the design engineer, test engineer or an automatic test vector generation program, which confirm that the device is operating properly after programming. For instance, structured vectors can ensure that nothing can happen in the device to prevent the 10-bit counter from operating correctly. An exhaustive set of device vectors will assure that no undetectable faults may occur.
<b>Sumcheck</b>	A 4- or 8-digit hexadecimal number that, when compared to the original data, allows you to verify that a copy of the data matches the original data. Memory devices have 8-digit sumchecks and logic devices have 4-digit sumchecks. For devices in a set, you can calculate the individual sumcheck of the device and the sumcheck of the entire set.
<b>Terminal Emulator</b>	A program to enable a PC or other computer to act as an ASCII terminal. Allows a PC to be used to communicate with a programmer in terminal mode or with a mainframe.
<b>Terminal Mode</b>	One of the programmer's three operating modes. The programmer is controlled from either a dedicated terminal or from a workstation running a terminal emulation package. Device data files can be stored on the programmer's disk (and on the workstation).
<b>Test Vector</b>	Test vectors functionally test the device, using structured test vectors stored in memory or in a disk file.
<b>Test Vector Stretching</b>	Conversion of DIP test vectors to equivalent PLCC test vectors by adding don't care vector characters into the string to correspond with the PLCC's dead pins.

<b>Total set size</b>	Used during serial set programming, this value specifies how many devices are in a set.
<b>Translate DIP/LCC Vectors</b>	See JEDEC I/O translate DIP/LCC Vectors
<b>Translation Formats</b>	A form of transmission protocol, these formats are used when transferring data between the programmer and a host computer. The different formats represent different ways of encoding the device data in a data file. The data file could contain the fuse pattern for a logic device or the data for a memory device.
<b>Transmit Pacing</b>	The number of milliseconds the programmer will insert as a time-delay between characters transmitted to the host computer during uploading. The time delay is specified in tenths of milliseconds.
<b>Transparent Mode</b>	One of the programmer's three operating modes. The programmer is controlled from either a dedicated terminal or from a workstation running a terminal emulation package. In addition, the Remote port is connected to a host computer or file server. The programmer performs all of its programmer operations, and can act as a transparent link between the terminal/workstation and the host, eliminating the need for a switch box. Thus, you can control both the programmer and the host from the same terminal/workstation. With one exception, all key strokes entered on the terminal will be passed directly to the host. The exception is the Escape command.
<b>TSOP</b>	An acronym for Thin Small Outline Package, a type of device package. A rectangular device with gull-wing leads on the short sides. Also called TSOP I.
<b>Underblow</b>	A condition in which fuses that should have been blown or programmed were not.
<b>Underblown Fuse</b>	A fuse that did not disconnect as per manufacturer's specifications. These fuses may test properly but tend to be more prone to grow back when in circuit, rendering the PLD useless.
<b>Universal Pin Driver</b>	A pin driver with the ability to supply power and ground to every pin. With Universal Pin Drivers, you can program and test devices without having to use pin out adapters and characterizers.
<b>Universal PLD Programmer</b>	A programmer that can apply power, ground, and any programming pulse required to program any fuse technology device.
<b>Upload Data</b>	A file transfer operation that involves sending data from the programmer to a host.
<b>Upload Host Command</b>	A command that is sent from the programmer to the host during an upload. The command tells the host what to do with the incoming data.
<b>Upload Wait</b>	The length of time the programmer will wait before it begins sending data to the host computer after the host upload command is sent.



<b>User Data Size</b>	The hexadecimal number of bytes of a data block to use for a device operation. Normally, this value is equal to the device size. During serial set operations, this value works with Total Set Size to determine the total amount of bytes to program into a set of devices.
<b>User memory</b>	The workspace used during device operations. It can be either internal RAM or a disk file. Normally, RAM is used for small, quick device operations, such as programming a single device, while disk is used for larger device operations, such as serial set programming.
<b>User Menu Data</b>	The information you see when you look at the programmer screen. It includes such items as the dialog window, reminder bar, and message bar.
<b>User Menu Port</b>	The port the user menu data is sent to. You can re-direct user menu data to either the Terminal port of the Remote port.
<b>User RAM</b>	The RAM in the programmer. User RAM can be used as a source/destination for an operation. Several operations use User RAM as a temporary storage buffer, overwriting any data that may have been there previously.
<b>Verify Device</b>	A device operation that compares data in a programmed device with data in RAM or in a disk file. With logic devices, verifying can also include functional testing. Verify is an automatic part of the program operation, but additional verify operations can provide useful information about any errors.
<b>Verify Pass</b>	A verify pass is a trip through a device at a specified Vcc to see if the device programmed properly. The pass is usually done once at 5V. The pass can also be done twice, with the first pass at 5.5V and the second pass at 4.5V.
<b>Waveforms</b>	Images of the programming pulses that program a device. Usually created by programmer manufacturers and submitted to device manufacturers as part of the approval process and to record the correct programming spec for a specific device.
<b>Wildcard</b>	Used when entering filenames, a wild card represents one or more characters in a filename. For example, 27*.dat represents both 27512.dat and 27128.dat.
<b>Workstation</b>	A PC, or other micro computer, used for local control of the programmer. You must use terminal emulation software, such as Data I/O's HiTerm, to allow the programmer and the PC (or other micro) to communicate. The programmer is designed to be compatible with all popular design workstations, including both DOS and UNIX-based workstations.
<b>Yield</b>	The percentage of successfully programmed devices.

**Yield Tally**

The yield tally function keeps track of the programming statistics for the last 16 types of devices programmed. The following statistics are kept for each device type: the manufacturer name and part number, the family/pinout code, the number of devices attempted, the number of devices that programmed successfully, the number of devices that failed non-blank test or illegal bit check, the number of devices that failed to verify, the number of devices that could not be programmed because they contained bits that required more programming pulses than were specified, and, for logic devices only, the number of devices that failed structured vector test.

**ZIF Socket**

An acronym for Zero Insertion Force. A socket in which the device can be dropped in and engaged via a lever.



# ***I**ndex*

---

## **A**

- Aborting a data transfer, 7-4
- Aborting an operation
  - See* Canceling an operation
- AC receptacle, 1-4
- Accessories
  - See* Options
- Accessory package
  - See also* Options
- Action symbol
  - description of, 3-2
  - See also* Interface
- Algorithm Disk
  - duplicating, 5-82
  - inserting, 2-17
- Algorithm source
  - default setting, 5-3
  - Edit programming parameters, 5-31
- Algorithm/System Disk
  - description of, 2-2
  - duplicating, 2-18, 2-28
  - inserting, 2-28
- Antistatic wrist strap
  - connecting to, 2-21
  - minimum resistance, 2-21
- AutoBaud
  - baud rates, 2-25
  - executing, 2-25, 3-5
  - explanation of, 2-25
  - User Menu Port, 2-25
  - when to use, 2-25

- B**
  - Backup disk. making
    - See* Duplicate Disk
  - Bases
    - DIP packages, 2-20
    - conductive pad, 2-39
    - installing, 2-20
    - introduction to, 2-20
    - MatchBooks, 2-20
    - PLCC packages, 2-20
    - removing, 2-20
  - Baud Rates
    - edit serial port configuration, 5-35
  - Blank Check
    - default setting, 5-3
    - device checks, 5-56
    - edit programming parameters, 5-32
    - overview, 5-62
    - program logic device, 5-18
    - program memory device, 5-21
  - Block Size
    - Complement Data, 5-74
    - Fill Data, 5-75
    - Move Data, 5-74
    - Swap Data, 5-76
  - Booting UniSite, 2-21, 2-24, 2-43, 3-5
  - Box (menu map symbol), 5-1
  - Bulletin Board Service, xxvi
  - Byte swap
    - See* Odd/Even Byte Swap
- C**
  - Cables
    - 25-pin to 25-pin, 2-14
    - 25-pin to 9-pin, 2-14
    - building your own, 2-14
    - electromagnetic interference, 2-6, 2-9, 2-12
    - pinout diagram, 2-14
    - shielding, 2-6, 2-9, 2-12
  - Calculate sumcheck
    - logic device, 5-56
    - memory device, 5-57
    - overview, 5-56
  - Calibration, 1-6
    - at power up, 1-6
    - at self-test, 1-6
    - checking, 1-6
  - Canadian Standards Association, 1-6
  - Canceling an operation, 3-5
  - Carrying case, 1-8
  - Changing the line voltage, 2-40
  - Changing your address, xxvii

- Clear Vectors, 5-70
- Command
  - executing, 3-3, 3-5
  - selecting, 3-3
- Command window, 3-2
- Communication parameters
  - changing, 5-36
  - See* Setup and installation
- Compare Data
  - overview, 5-92
- Compare Device Electronic Signature, 5-59
- Compare Electronic ID
  - default setting, 5-3
  - device checks, 5-56
  - edit programming parameters, 5-32
  - load memory device, 5-16
  - overview, 5-59
  - program memory device, 5-21
  - verify memory device, 5-26
- Compatibility, software version, 2-19
- Compatible terminals
  - list of, 5-45
  - See also* Terminal type
- Compensated Vector Test
  - edit programming parameters, 5-34
- Complement Data
  - overview, 5-74
- Complement Programmer Memory, 5-74
- Complex Parameter Screens
  - See also* All Parameters
- Computer Remote Control
  - command summary, 6-6 – 6-7
  - default settings, 6-5
  - entering, 5-87
  - entering CRC mode, 6-2
  - exiting, 5-87, 6-4
  - halting an operation, 6-4
  - overview, 5-87
  - power-up CRC mode, 6-3
  - software driver, 6-1
  - system setup, 6-2
- Configuration Inquiry, overview, 2-1, 5-28
- Configuring peripherals, 2-10, 2-13
- Configuring the programmer, 2-1
- Configuring UniSite
  - See* Setup and Installation
- Contents of package, 1-2
- Context-sensitive Help, 3-4
- Continuity Check
  - default setting, 5-3
  - edit programming parameters, 5-33
- Copy Data File, 5-81

- Copy Disk
  - See* Duplicate Disk
- Copy File, overview, 5-81
- Cross Programming, overview, 5-11
- Cursor control, 3-3
- Custom Menu Algorithms, 5-50
  - Add, 5-52
  - Create, 5-50
  - Delete, 5-53
  - Update, 5-54
  - View, 5-53
- Customer Support, xxiii – xxiv

## D

- D (CRC Command), 6-7
- Data Bits
  - Edit Serial Port Configuration, 5-35
- Data Destination
  - default setting, 5-3
- Data File Directory, 5-76
- Data I/O
  - addresses, xxiii – xxiv
  - Bulletin Board Service, xxvi
  - contacting via BBS, xxvi
  - contacting via telephone, xxv
- Data location
  - Compare Data, 5-93
- Data representation
  - Edit Fuse Map, 5-66
- Data source
  - default setting, 5-3
- Data Translation Format
  - default setting, 5-4
  - selecting, 5-94
  - session on, 4-23
- Data Word Width
  - default setting, 5-3
  - Edit Programming Parameters, 5-31
  - Illegal Bit Check Memory Device, 5-61
  - Load Memory Device, 5-14
  - Program Memory Device, 5-19
  - Sumcheck Memory Device, 5-58
  - Verify Memory Device, 5-25
- Delete File
  - Keep Current, 5-48
  - See* Purge File

- Destination
  - See* Data Destination
  - Download Data, 5-89
  - Edit Communication Parameters, 5-36
  - Edit Programming Parameters, 5-30
  - Input from Disk, 5-95
  - Load Logic Device, 5-13
  - Load Memory Device, 5-14
  - Output Logic Data, 5-98
  - Output Memory Data, 5-98
  - Upload Data, 5-91
- Device
  - inserting DIPs, 4-17
  - inserting PLCC or LCC into a MatchBook, 2-36
  - installation, 2-34
- Device Begin Address
  - default setting, 5-3
  - Edit Programming Parameters, 5-32
  - Load Memory Device, 5-15
  - Program Memory Device, 5-21
  - Verify Memory Device, 5-26
- Device Block Size
  - default setting, 5-3
  - Load Memory Device, 5-15
  - Program Memory Device, 5-21
  - Verify Memory Device, 5-26, 5-32
- Device Checks
  - overview, 5-56
- Device configure
  - overview, 5-65
- Device List on a Disk, 2-2
- Device support, 2-2
- Device, Selecting a
  - See* Select Device
- Dialog Window, 3-3
- DIP devices
  - See* Device
- DIPs
  - installing, 2-34
- Disk drive, second drive, 1-7
- Disks
  - copying, 5-82
  - inserting, 2-17
  - write protecting, 2-18
- DOS
  - Duplicate Disk, 5-82
- Download Data
  - overview, 4-26 – 4-27, 4-29, 5-89
- Download Echoing
  - Edit Communication Parameters, 5-37
- Download End-of-file Delimiter
  - Edit Communication Parameters, 5-39



- Download Host Command
  - Compare Data, 5-94
  - default setting, 5-4, 5-90
  - Download Data, 5-90
  - Edit Communication Parameters, 5-39
- Download: Use End-of-file Delimiter
  - Edit Communication Parameters, 5-38
- Drive
  - A, 1-3
  - B, 1-3
- Duplicate Disk
  - after first use, 2-18
  - after system update, 2-28
  - overview, 5-82
  - suggestions, 5-82

## E

- Edit Address Offset
  - Edit Memory Data, 5-71
- Edit Begin Address
  - Edit Memory Data, 5-71
- Edit Begin Vector
  - Edit Test Vectors, 5-68
- Edit Data
  - See also* Edit Fuse Map
  - See also* Edit Logic Data
  - See also* Edit Memory Data
  - logic devices, 5-66
  - memory devices, 5-71
  - overview, 5-66
  - session on, 4-32
- Edit Data Word Width
  - Edit Memory Data, 5-71
- Edit Fuse Map
  - editor commands, 5-66
  - exiting, 5-67
  - restoring edits, 5-67
- Edit Logic Data
  - See* Edit Fuse Map
- Edit Memory Data
  - Editor Commands, 5-72
  - overview, 5-71
- Edit Programmer Memory, 5-71
- Edit Programming Parameters, 5-30
- Edit Vector
  - commands, 5-69
  - overview, 5-67
  - restoring edits, 5-69
  - saving edits, 5-69
- EE Bulk Erase
  - default setting, 5-3
  - Device Checks, 5-56

- Electrical shock
  - avoiding, 2-21
- Electromagnetic Interference
  - minimizing, 2-6, 2-9, 2-12
- Electronic ID
  - See Compare Electronic ID
- Electronic mail
  - See Technical assistance
- Electronically Erase Device, 5-62
- Electrostatic Discharge, 1-6, 2-21
- Enable CTS/DTR
  - Edit Serial Port Configuration, 5-35
- Enable Terminal Beeps
  - Edit Interface Parameters, 5-40
- Enable Yield Tally
  - Program Logic Device, 5-18
  - Program Memory Device, 5-22
- End user registration, xxvii
- End User Registration Card, 2-17
- Enhanced Security Fuse Capability
  - overview, 5-22
- Environmental Specifications
  - See Specifications
- Erase EE Device
  - Edit Programming Parameters, 5-33
  - Program Memory Device, 5-22
- Error Messages
  - See Messages
- Extended Algorithm
  - See Algorithm Source
- Extended algorithms, selecting, 5-10
- Extended CRC Commands, list of commands, 6-8

**F**

- Factory Default Settings
  - listing of, 5-3
  - restoring, 5-3
- Family Code
  - default setting, 5-3
  - selecting, 5-5
- Features
  - front panel, 1-3
  - pack panel, 1-4
- Fields, special parameters, 5-10
- File Menu, 4-21
- File Operations, overview, 5-76
- Filename
  - Compare Data, 5-93
  - default setting, 5-4
  - Download Data, 5-89
  - Edit Fuse Map, 5-66
  - Edit Memory Data, 5-71

- Edit Programming Parameters, 5-30
- Edit Test Vectors, 5-68
- Illegal Bit Check Logic Device, 5-60
- Input from Disk, 5-95
- Load File, 5-77
- Load Logic Device, 5-13
- Load Memory Device, 5-14
- Output Logic Data, 5-98
- Output Memory Data, 5-97
- Output to Disk, 5-96
- Program Logic Device, 5-17
- Program Memory Device, 5-19
- Save File, 5-78
- Sumcheck Logic Device, 5-57
- Sumcheck Memory Device, 5-57
- Under/Over-Blow, 5-63
- Upload Data, 5-91
- Verify Logic Device, 5-23
- Verify Memory Device, 5-25
- Wildcards, 5-81
- Fill Data
  - Edit Communication Parameters, 5-37
- Fill Fuse Map, overview, 5-70
- Fill Memory Option
  - Edit Communication Parameters, 5-37
  - overview, 5-75
- Fill Programmer Fuse Map, 5-70
- Fill Programmer Memory, 5-75
- Fill Variable
  - Fill Data, 5-75
- For Member X of Y
  - Sumcheck Memory Device, 5-58
- Format Disk, overview, 5-84
- Format Select, 5-94
- From Memory Address
  - Move Data, 5-74
- FSM, 8-1
- Functional specifications, 1-5
- Fuse, replacing, 2-41
- Fuse ratings, 1-5

## G

- Generic PLD and Cross Programming, 5-11
- Ground connector, 1-3, 2-21
- Grounding
  - See* Antistatic wrist strap

**H**

- Halting an operation
  - in CRC, 6-4
- Hardware Handshake
  - See* Handshaking
- Hardware Handshaking, 7-3 – 7-4
- Help
  - See* Online Help
- Help (technical)
  - See* Technical assistance
- High Speed Download
  - benefits, 2-29
  - default setting, 5-4
  - Edit Communication Parameters, 5-37
  - power-up defaults, 2-31
  - requirements, 5-37
  - setting up, 2-29
- High Speed Logic Drivers
  - Edit Programming Parameters, 5-34
- HiTerm, 2-2
  - capabilities, 2-7
  - High Speed Download, 2-7, 2-29
  - installing, 2-7
- Humidity, 1-5

**I**

- I/O Addr Offset
  - Compare Data, 5-93
  - Edit Communication Parameters, 5-36
- I/O Address Offset
  - default setting, 5-4
  - Download Data, 5-90
  - Input from Disk, 5-95
  - Output To Disk, 5-97
  - Upload Data, 5-91
- I/O Timeout
  - default setting, 5-4
  - Edit Communication Parameters, 5-36
- I/O Translation Format
  - Compare Data, 5-93
  - See* Data Translation Format
  - Download Data, 5-89
  - Edit Communication Parameters, 5-36
  - Input from Disk, 5-95
  - Output To Disk, 5-97
  - Upload Data, 5-91

- Illegal Bit Check
  - default setting, 5-4
  - Device Checks, 5-56
  - Edit Programming Parameters, 5-32
  - logic device, 5-60
  - memory device, 5-61
  - overview, 5-60
  - Program Logic Device, 5-18
  - Program Memory Device, 5-21
- Individual sumcheck
  - Sumcheck Memory Device, 5-58
- Input From Disk, overview, 5-95
- Inserting DIP Devices, 4-17
- Installation
  - module, 2-3
  - See Setup and Installation
- Installing Software
  - See Update
- Instrument control code
  - default setting, 5-4
  - Edit Communication Parameters, 5-37
- Instrument control codes, 7-3
- Interface
  - action symbol, 3-2
  - areas of the screen, 3-2
  - changing a parameter, 4-19
  - command window, 3-2
  - controlling the cursor, 3-3
  - dialog window, 3-3
  - entering a parameter, 4-19
  - message bar, 3-2
  - online Help, 3-3
  - overview, 3-2
  - reminder bar, 3-3
  - selecting a command, 3-3
  - status window, 3-2
- Interface parameters, changing, 5-40

## J

- JEDEC I/O Translate DIP/LCC Vectors
  - Edit Communication Parameters, 5-38
  - See Translate DIP/LCC option
- Job File
  - accessing from Main Menu, 5-40
  - overview, 5-85
  - playing back, 5-86
  - recording, 3-5, 5-85
  - system software updates, 5-85 – 5-86
  - tips on recording, 5-85
  - using with new software, 5-85 – 5-86

- 
- K**
- Keep Current, 5-46
    - Delete File, 5-48
    - Purge Files, 5-49
    - Replace/Restore, 5-46
    - View Files, 5-46
  - Keep Current Subscription Service, xxvii, 1-6
  - Key functions, 3-5
- L**
- LEDs, definition of, 1-3
  - Line Voltage
    - See* Voltage
  - Load Data
    - from a device, 5-13
    - from a device, session on, 4-16
    - from a host, session on, 4-28
    - from a master device, 4-15
    - from a PC, session on, 4-24
    - from a port, 4-15
    - from disk, 4-15
    - overview, 4-2
  - Load Data File, 5-77
  - Load Data from Disk, session on, 4-20
  - Load Device, overview, 4-2, 5-13
  - Load File, overview, 5-77
  - Load Logic Device, 5-13
  - Load Memory Device, 5-14
  - Logic Device Illegal Bit Check, 5-60
  - Logic Verification
    - default setting, 5-4
    - Edit Programming Parameters, 5-30
    - Program Logic Device, 5-18
    - Verify Logic Device, 5-24
- M**
- Main Menu Job Files
    - Edit Interface Parameters, 5-40
  - Maintenance, 1-6, 2-40
  - Maintenance Manual
    - description of, 1-8
    - See also* Options
  - Manufacturer
    - default setting, 5-4
    - selecting, 5-5 – 5-6
  - Mass Storage command, 5-55
  - Mass Storage Module (MSM), 2-32
  - MatchBooks
    - installing, 2-34
    - removing, 2-39

- Memory
  - See* System memory
- Memory Address
  - Complement Data, 5-74
- Memory Begin Address
  - Compare Data, 5-93
  - default setting, 5-4
  - Download Data, 5-90
  - Edit Programming Parameters, 5-32
  - Fill Data, 5-75
  - Input from Disk, 5-95
  - Load File, 5-77
  - Load Memory Device, 5-15
  - Output Memory Data, 5-98
  - Output To Disk, 5-97
  - Program Memory Device, 5-21
  - Save File, 5-78
  - Sumcheck Memory Device, 5-57
  - Swap Data, 5-75
  - Upload Data, 5-92
  - Verify Memory Device, 5-26
- Memory Device Illegal Bit Check, 5-61
- Menu Maps
  - description, 5-1
  - using, 5-1
- Message Bar, 3-2
- Messages, display of, 3-2
- Module installation, 2-3
- Modules
  - and DIP packages, 2-3
  - inserting devices, 4-17
  - introduction to, 2-3
  - and MatchBooks, 2-3
  - and PLCC/LCC packages, 2-3
- Move Data, overview, 5-74
- Move Programmer Memory Data, 5-74
- MSM, 5-55

## N

- New Software
  - See* Update
- Next Device
  - Illegal Bit Check Memory Device, 5-61
  - Load Memory Device, 5-14
  - Program Memory Device, 5-20
  - Verify Memory Device, 5-25
- Next Operation Begins At
  - Illegal Bit Check Memory Device, 5-61
  - Load Memory Device, 5-15
  - Program Memory Device, 5-20
  - Sumcheck Memory Device, 5-58
  - Verify Memory Device, 5-25

- Number of lines between form feeds
  - Output Logic Data, 5-98
  - Output Memory Data, 5-98
- Number of nulls
  - Edit Communication Parameters, 5-37
- Number of vectors
  - Output Logic Data, 5-98

## O

- Odd/Even Byte Swap
  - default setting, 5-4
  - Edit Programming Parameters, 5-33
  - Load Memory Device, 5-16
  - Program Memory Device, 5-22
  - Verify Memory Device, 5-26
- Online Help
  - accessing, 3-3, 3-5
  - Context-sensitive help, 3-4
  - exiting, 3-5
  - for device messages, 3-4
  - for system messages, 3-4
  - General Help, 3-3
  - Key Listing, 3-3
  - Reminder Bar, 3-4
- Operating Modes
  - Transparent mode, 3-5, 5-101
- Operating Specifications
  - See* Specifications
- Operating Voltage
  - See* Voltage
- Options, 1-6
  - descriptions of, 1-6 – 1-8
  - ordering information, 1-6
- Output Filename
  - Output to Disk, 5-96
- Output Logic Data, 5-98
  - See* Serial Output
- Output Memory Data, 5-97
  - See* Serial Output
- Output Record Size
  - Edit Communication Parameters, 5-37
- Output to Disk
  - overview, 5-96
- Over Blow
  - Device Checks, 5-56
  - See* Under/Over-Blow



**P**

- Package Contents, 1-2
- Parameter Screens
  - See also* All Parameters
  - See* Non-default Parameters
- Parameters
  - changing, 4-19
  - CRC defaults, 6-5
  - description of, 4-18
  - entering, 4-19
  - overview, 4-18
  - power-up defaults, 5-3
  - setting, 4-19
- Parity
  - Edit Serial Port Configuration, 5-35
- Part Number
  - default setting, 5-4
- PC, connecting to, 2-6
- Physical specifications, 1-5
- Pin driver, description of, 1-8
- Pinout Code
  - default setting, 5-4
  - selecting, 5-5
- PinSite, 1-7
- Power cord, connecting, 2-21
- Power LED, 1-4
- Power-on screen, 2-26
- Power-on CRC mode
  - default setting, 5-4
  - Edit Interface Parameters, 5-40
- Power requirements, 1-5
  - frequency range, 1-5
  - operating voltages, 1-5
  - power consumption, 1-5
- Power switch, 1-4
- Powerup defaults
  - changing, 2-31
  - high speed download, 2-31
  - overview, 2-31
  - Restore System Parameters, 5-42
  - Save System Parameters, 5-41
- Powerup LEDs, 2-22
- Powering up, 2-21 – 2-22, 2-43
- Program device
  - See also* Quick Copy
  - overview, 4-2, 5-17
  - session on, 4-35
- Program logic device, 5-17
- Program memory device, 5-19

- Program security fuse
  - default setting, 5-4
  - Edit Programming Parameters, 5-33
  - Program Logic Device, 5-18
  - Program Memory Device, 5-20
- Program signature
  - Program Memory Device, 5-20
- Programmer Fuse Map Edit, 5-66
- Programmer ID, viewing, 5-42
- Programmer Test Vector Edit, 5-67
- Programming session on, 4-35
- Programming concepts, 4-2
- Programming parameters, overview, 5-30
- PSM, 8-1
- Purge Data File, 5-79
- Purge File, overview, 5-79
- Purge Files, Keep Current, 5-49

**Q**

- Quick Copy, 5-12
  - SetSite, 5-12
- Quick Start, 3-1

**R**

- RAM
  - See* System memory
- RAM device selection
  - Edit Programming Parameters , 5-34
- RAM upgrades, ordering information, 1-7
- Registration, xxvii
- Reject option
  - default setting, 5-4
  - Edit Programming Parameters, 5-30
  - Program Logic Device, 5-19
  - Program Memory Device, 5-22
- Reminder bar
  - command screens, 3-3
  - help screens, 3-4
- Remote Control
  - See* Computer Remote Control
- Remote LED, 1-4
- Remote Off code
  - default setting, 5-4
  - Edit Interface Parameters, 5-40
- Remote On code
  - default setting, 5-4
  - Edit Interface Parameters, 5-40
- Remote port, 1-4
  - default setting, 5-4
- Rename data file, 5-80
- Rename file
  - overview, 5-80

Repair information  
    *See also* Customer Support  
Repair Service  
    information, xxvii  
    ordering, xxvii  
Restore System Parameters, 5-42

## S

Safety  
    Canadian Standards Association, 1-6  
    CE, 1-6  
    specifications, 1-6  
    Underwriters Laboratories, 1-6  
    Verband Deutscher Electrotechniker, 1-6  
Save Data File, 5-78  
Save File  
    overview, 5-78  
Save System Parameters, 5-41  
Screen Format, 3-2  
second disk drive, 1-7  
Security Fuse Data  
    default setting, 5-4  
    edit programming parameters, 5-30  
    *See also* Enhanced Security Fuse Capability  
    general information, 5-22  
    program logic device, 5-17  
    program memory device, 5-20  
Select device  
    by manufacturer and part number, 5-5  
    overview, 4-2, 4-6, 4-11, 5-5  
    tutorial, 4-6, 4-11  
Selecting a command, 3-3  
Selecting a translation format  
    *See* Format Select  
Self-test, calibration, 1-6  
Self-test LED, 1-4  
Self-test  
    continuous, 5-87  
    halting, 5-87  
    interpreting, 2-22, 5-88  
    on power-up, 2-22, 5-88  
    one-pass, 5-87  
    overview, 5-87  
    running, 5-87  
Serial cables  
    *See* Cables  
Serial output  
    logic data, 5-98  
    memory data, 5-97  
    overview, 5-97  
Serial port configuration  
    changing, 5-35

- Serial ports
  - See* Remote port
  - See* Terminal port
- Serial Vector Test
  - edit programming parameters, 5-33
- Set Auto-increment
  - Load Memory Device, 5-16
  - Program Memory Device, 5-21
  - Verify Memory Device, 5-26
- Set Operations
  - sumchecking, 5-58
- Set Sumcheck
  - Sumcheck Memory Device, 5-58
- SetSite, 1-7
- Setting up the programmer
  - overview, 2-1
- Setup
  - See* Setup and Installation
- Setup and installation
  - connecting to a host, 2-9
  - connecting to a PC, 2-6
  - connecting to a terminal, 2-11
  - overview, 2-1
- Shaded box (menu map symbol), 5-1
- Site48HS, 1-7
- SmartPort, 2-13
- Software data protection
  - program memory device, 5-20
- Software handshake
  - See* Handshaking
- Software License Agreement, 2-1
- Software updates
  - See also* Update
- Software version compatibility, 2-19
- Source
  - Compare Data, 5-93
  - Download Data, 5-89
  - Edit Communication Parameters, 5-36
  - Edit Fuse Map, 5-66
  - Edit Memory Data, 5-71
  - Edit Programming Parameters, 5-30
  - Edit Test Vectors, 5-67
  - Illegal Bit Check Logic Device, 5-60
  - Illegal Bit Check Memory Device, 5-61
  - Output Logic Data, 5-98
  - Output Memory Data, 5-97
  - Output to Disk, 5-96
  - Program Logic Device, 5-17
  - Program Memory Device, 5-19
  - Sumcheck Logic Device, 5-57
  - Sumcheck Memory Device, 5-57
  - Under/Over-Blow, 5-63

- Upload Data, 5-91
- Verify Logic Device, 5-23
- Verify Memory Device, 5-25
- Special parameter fields, 5-10
- Specifications
  - CE, 1-6
  - environmental, 1-5
  - functional, 1-5
  - physical, 1-5
  - power requirements, 1-5
  - safety, 1-6
- Starting vector number
  - Output Logic Data, 5-98
- Status indicators, 1-4
- Status window, 3-2
- Stop Bits
  - Edit Serial Port Configuration, 5-35
- Sumcheck Display
  - device checks, 5-56
  - logic devices, 5-56
  - memory devices, 5-57
  - of a set of devices, 5-58
  - overview, 5-56
- Sun, connecting to, 2-9
- Support
  - See* Customer Support
- Swap Data, overview, 5-75
- Swap mode
  - Swap Data, 5-75
- Swap Programmer Memory, 5-75
- System Diagnostic Tests
  - See* Self-Test
- System Disk
  - duplicating, 2-18, 5-82
  - inserting, 2-17
- System memory
  - amount installed, 3-2
  - optional, 1-7
  - standard, 1-5
- System Messages
  - See* Messages
- System Software
  - update, 2-28

## T

- T-box (menu map symbol), 5-1
- Technical assistance, xxv
- Technical Specifications
  - See* Specifications
- Technical support
  - See* Customer Support

- Terminal
  - connecting to, 2-11
  - support, 1-5
- Terminal Emulation Software
  - HiTerm, 2-6 – 2-7
- Terminal LED, 1-4
- Terminal port, 1-4
  - default setting, 5-4
- Terminal Support
  - See* Terminal Type
- Terminal Type
  - changing, 2-26, 5-45
  - compatible terminals, 2-12, 5-45
  - current setting, 2-26, 5-45
  - default setting, 2-27, 5-4, 5-45
  - selecting, 5-45
  - selecting new, 2-26
- To Memory Address
  - Move Data, 5-74
- Total Set Size
  - Illegal Bit Check Memory Device, 5-61
  - Load Memory Device, 5-15
  - Program Memory Device, 5-20
  - Sumcheck Memory Device, 5-58
  - Verify Memory Device, 5-25
- Transfer data
  - overview, 5-89
- Translate DIP/LCC option
  - default setting, 5-4
- Translation formats
  - Introduction, 7-1
  - Selecting, 7-1
- Transmit pacing
  - default setting, 5-4, 5-39
  - Edit Communication parameters, 5-36
  - overview, 5-39
  - with HiTerm, 5-39
  - with PROCOMM, 5-39
  - with VTERM, 5-39
- Transparent mode
  - configuring for, 2-9, 2-11
  - entering, 2-9, 2-11, 3-5, 5-101
  - ESC key, 5-101
  - exiting, 2-9, 2-11, 5-101
  - overview, 2-9, 2-11, 5-101
- Troubleshooting, 2-24
- Tutorial, 4-1
- Typographic conventions, xxviii

**U**

- Under/Over-Blow
  - Device Checks, 5-56
- Underblow/Overblow
  - commands, 5-64
  - overview, 5-63
- Underwriters Laboratories, 1-6
- UniSite
  - Actel Gang Module, 1-8
  - back panel, 1-4
  - calibration, 1-6
  - description of, 1-1
  - drives, 1-3
  - external features, 1-3
  - front panel features, 1-3
  - fuse, 2-41
  - gang programming, 1-7 – 1-8
  - introduction to, 1-1
  - Keep Current Subscription Service, 1-6
  - and LCC package types, 1-7
  - Maintenance Manual, 1-8
  - module, 2-3
  - options, 1-6
  - package contents, 1-2
  - and PGA package types, 1-7
  - Pin driver, 1-8
  - PinSite, 1-7
  - and PLCC package types, 1-7
  - and QFP package types, 1-7
  - RAM, 1-5
  - RAM upgrades, 1-7
  - and SDIP package types, 1-7
  - second disk drive, 1-7
  - set programming, 1-7
  - SetSite, 1-7
  - Site48HS, 1-7
  - and SOIC package types, 1-7
  - specifications, 1-5
  - status indicators, 1-4
  - USM-340 Module, 1-8
  - voltage changing, 2-40
- UniSite options
  - See* Options
- Unlocking Software
  - See* Update
- Update
  - after the update, 2-28
  - verifying, 5-55
- Updates
  - early updates, 1-6
  - Keep Current Subscription Service, xxvii, 1-10
  - ordering, 1-6

- Upload data, 5-91
- Upload destination
  - default setting, 5-4, 5-91
- Upload End-of-file Delimiter
  - Edit Communication Parameters, 5-38
- Upload Host Command
  - default setting, 5-4, 5-92
  - Edit Communication Parameters, 5-39
  - Upload Data, 5-92
- Upload Record Size
  - default setting, 5-4
- Upload Wait
  - default setting, 5-4
  - Edit Communication Parameters, 5-36
- Upload: Use End-of-file Delimiter
  - Edit Communication Parameters, 5-38
- User Data Size
  - Compare Data, 5-94
  - default setting, 5-4
  - Download Data, 5-90
  - Edit Programming Parameters, 5-32
  - Illegal Bit Check Memory Device, 5-61
  - Input from Disk, 5-96
  - Load File, 5-77
  - Load Memory Device, 5-15
  - Output Memory Data, 5-98
  - Output To Disk, 5-97
  - Program Memory Device, 5-20
  - Save File, 5-78
  - Sumcheck Memory Device, 5-58
  - Upload Data, 5-92
  - Verify Memory Device, 5-25
- User Interface
  - See* Interface
- User Menu Port, 6-3
  - AutoBaud, 2-25
  - changing, 2-29
  - default setting, 5-4
  - Edit Communication Parameters, 5-38
  - High Speed Download, 2-29
  - overview, 2-30
- User RAM
  - overwriting, 5-3
- Utility disk, 2-2

## V

- VAX
  - connecting to, 2-9
- Vector Edit
  - See also* Clear Vectors
  - See* Edit Vector



- Verify Data Format
  - default setting, 5-4
  - Edit Programming Parameters, 5-31
- Verify Device
  - against serial port data, 5-92
  - overview, 4-2, 5-23
- Verify Logic Device, 5-23
- Verify Memory Device, 5-24
- Verify Passes
  - default setting, 5-4
  - Edit Programming Parameters, 5-31
  - Program Logic Device, 5-19
  - Program Memory Device, 5-22
  - Verify Logic Device, 5-24
  - Verify Memory Device, 5-26
- Verifying
  - session on, 4-38
- Version, 2-19
- View Device Family/pinout Code, 1-10
- View Directory
  - overview, 5-76
- Voltage
  - acceptable range, 1-5
  - changing the, 2-40
  - See also* Frequencies

## W

- Warranty
  - information and service, xxvi
- Word Width
  - See* Data Word Width
- Wrist strap
  - See* Antistatic wrist strap

## Y

- Yield Tally
  - default setting, 5-3
  - Edit Programming Parameters, 5-32
  - erasing statistics, 5-99
  - overview, 5-99
  - statistics kept, 5-99